Instituto Politécnico de Tomar

DESIGNING 3D SCENARIOS AND INTERACTION TASKS FOR IMMERSIVE ENVIRONMENTS

Beril Yalçınkaya
21461

A Thesis Report submitted in order to obtain the degree of Master Science in Informatics Engineering

Supervisors
Professor Gabriel Pires
Professor Luis Almeida

2020

ACKNOWLEDGEMENTS

I would like to express my appreciation for this chance that my supervisors gave me during the development of this project.

To my supervisors Professor Gabriel Pires, Professor Luis Almeida a big thank you for all the times you've showed your belief in me. I would like to thank Professor Ana Lopes for all meaningful help and support.

To Rodolfo Martins, Elio Lopes and Dylan Bicho for all the support and dedication shown as colleagues.

I would also like to thank all professors and friends from Polytechnic Institute of Tomar - IPT.

To Doctor Micael Couceiro, a very big thank you for always placing your trust and confidence in me and pushing me to be my better version.

Finally to my family, thank you for the endless support and for believing in me.

This work was supported by the RD project HTPDIR: HUMAN TRACKING AND PERCEPTION IN DYNAMIC IMMERSIVE ROOMS (reference POCI-01-0247-FEDER-017644). The activities were carried out in the Lab. VITA.IPT - Life Assisted by Intelligent Environments (Vida Assistida por ambientes Inteligentes) of the Polytechnic Institute of Tomar.

ABSTRACT

In the world of today, immersive reality such as virtual and mixed reality, is one of the most attractive research fields. Virtual Reality, also called VR, has a huge potential to be used in in scientific and educational domains by providing users with real-time interaction or manipulation. The key concept in immersive technologies to provide a high level of immersive sensation to the user, which is one of the main challenges in this field. Wearable technologies play a key role to enhance the immersive sensation and the degree of embodiment in virtual and mixed reality interaction tasks.

This project report presents an application study where the user interacts with virtual objects, such as grabbing objects, open or close doors and drawers while wearing a sensory cyberglove developed in our lab (Cyberglove-HT). Furthermore, it presents the development of a methodology that provides inertial measurement unit(IMU)-based gesture recognition.

The interaction tasks and 3D immersive scenarios were designed in Unity 3D. Additionally, we developed an inertial sensor-based gesture recognition by employing an Long short-term memory (LSTM) network. In order to distinguish the effect of wearable technologies in the user experience in immersive environments, we made an experimental study comparing the Cyberglove-HT to standard VR controllers (HTC Vive Controller). The quantitive and subjective results indicate that we were able to enhance the immersive sensation and self embodiment with the Cyberglove-HT. A publication resulted from this work [1] which has been developed in the framework of the R&D project Human Tracking and Perception in Dynamic Immersive Rooms (HTPDIR).

Key words: Technologies, wearable technologies, virtual reality, mixed

reality, immersive environments, immersive interactions, 3D scenario designing

Contents

Acknowledgements iii

Abstract iv

Acronyms ix

List of Figures x

List of Tables xiv

1 Introduction 2

1.1 Context and Motivation 2

1.2 Main Developments and Contributions 3

1.3 Document Structure 4

2 State of The Art, Background Methods and Materials 6

2.1 Concepts and Background Virtual Reality 7

2.1.1 Defining Virtual Reality 7

2.1.2 Historical Background of Virtual Reality 8

2.1.3 Main Trends and Applications: Challenges and Achievements . 10

2.1.4 Challenges in Virtual Reality 11

2.1.5 Related work: Interaction Tasks in Immersive Environments . . 11

2.2 Technologies: Head Mounting Devices, Tracking Systems, Controllers and Rendering Systems 16

2.2.1 Hardware Systems 16

2.3 Virtual Reality Software Frameworks 19

2.4 Classification Methods for Gesture Recognition 20

3 VR MODELS, INTERACTION AND MANIPULATION TASKS 22

3.1 Hardware System 22

3.1.1 Tracking system HTC Vive 22

3.1.2 CyberGlove-HT 23

3.2 Immersive Software Architecture 25

3.2.1 Software Framework 27

3.2.2 Communication and Data Parsing with CyberGlove-HT 28

3.2.3 Integration of 3D Models 31

3.2.4 Interaction Tasks 35

3.2.5 Integration of Kinect System 44

4 Gesture Recognition Based on Inertial Sensors 51

4.1 Matlab framework for IMU based gesture recognition 51

4.1.1 Data Collection 52

4.1.2 Classification Approach 54

4.1.3 Implementation of Gesture Recognition in Unity 58

5 Experimental Results **60**

5.1 Interaction Tasks in Immersive Environment 60

5.1.1 Experimental Design and Procedure 60

5.1.2 Results and Discussion 62

5.2 IMU-Based Gesture Recognition 65

6 Conclusion **67**

BIBLIOGRAPHY **68**

A **73**

ACRONYMS

ANN- Artificial Neural Networks
API- Application Programming Interface
AR- Augmented Reality
CPU- Central Processing Unit
CRT- Cathode Ray Tube
DoF- Degree of Freedom
HMD- Head Mounted Display
HT- Hand Tracker
HTPDIR- Human Tracking and Perception in Dynamic Immersive Rooms
IEEE- The Institute of Electrical and Electronics Engineers
IH- Inner Hand
IMU- Inertial Measurement Unit
IP- Internet Protocol
JSON- JavaScript Object Notation
LSTM- Long Short Term Memory
m- meter
MR- Mixed Reality
ms- milisecond
OBB- Oriented Bounding Box
OH- Outer Hand
PTT- Precision Position Tracker
RE- Real Environment
RNN- Recurrent Artificial Neural Networks
SDK- Software Development Kit
SLAM- Simultaneous Localization and Mapping
SVM- Support Vector Machine

UDP- User Datagram Protocol

UI- User Interface

VE- Virtual Environment

VR- Virtual Reality

Wi-Fi- Wireless Fidelity

List of Figures

1.1 Overview diagram of the immersive reality system 4

2.1 Milgram's reality-virtuality continuum *(taken from [2])* 7

2.2 Sensorama Simulation Machine 8

2.3 The Ultimate Display 9

2.4 The Virtual Wind Tunnel 9

2.5 Different type of grasps: Grasping a mug and a toy with power grip, a racket with cylindrical grasp and a bowl with an extension grip. 14

2.6 Hand-object interpenetration. 15

2.7 Google Cardboad is on the left and GearVR is on the right[1] 16

2.8 Oculus Quest VR is on the left and HTC VIVE Cosmos is on the right 17

2.9 Oculus Touch controller is on the left and HTC VIVE controller is on the right. 18

2.10 Workflow of a LSTM Network 21

3.1 The Immersive Room Setup 23

3.2 The HTC Vive components and TPCAST Wireless Adapter 24

3.3 The components of CyberGlove-HT 25

3.4 Detailed software diagram architecture and its interaction with the sensing devices. 26

3.5 Management/interaction of scripts responsible for implementing the actions. 27

3.6 The SteamVR Status Window 28

3.7 Setup of Immersive Room 28

3.8 The FlowChart of Communication and Data Parsing 29

3.9 The Hand Model 32

3.10 The house from behind perspective 33

3.11 The house from front perspective 33

3.12 The Piano Model 34

3.13 The Door Model 34

3.14 The Drawer Model 35

3.15 Systems coordinates of the hand and wrist in the virtual environment according to an egocentric view 36

3.16 The finger rotation is animated via flexible sensor data 37

3.17 The Flowchart of the Grabbing and Releasing Interaction 38

3.18 The hand is grabbing a cube 39

3.19 The hand is opening the door 40

3.20 The hand is pulling the handle of a drawer 41

3.21 Playing piano 42

3.22 The user is activating the Grabbing mode via the Interactive Menu . . 43

3.23 The flowchart of providing visual feedback 44

3.24 The Kinect quadrangular scenario 45

3.25 The Boundingbox of a human [3] 46

3.26 The flowchart of Human and Object Tracking Visualization 49

3.27 Virtual world built from the real world (image shown in projection screen). The Kinect sensor-based reconstruction system maps the physical elements as bounding boxes. 50

4.1 The Adopted Methodology of Imu-Based Gesture Recognition 52

4.2 The Quaternion Signals received from CyberGlove-HT while the user performs vertical swipe gesture 53

4.3 The sliding classification window 55

4.4 General classification approach applied to the hand motion (The hand figures are taken from [4]). 57

4.5 The Approach of Communication between Unity and MATLAB 59

4.6 The user swipes the menu up with swiping the hand vertically 59

5.1 Sub-tasks to open the door in VR environment. 61

5.2 Mean task-time performance of participants for each sub-task 63

5.3 a) Total mean task-time, and b) total mean length of the path described by the hand of participants while performing the virtual door opening global task, HTC Vive controller vs cyber-glove. 64

5.4 Mean length of the path described by the hand of participants, for each sub-task 64

5.5 Results of subjective questionnaires 65

5.6 The confusion matrix of the offline results 66

List of Tables

2.1 HTC VIVE Cosmos vs Oculus Quest VR Headset Comparison Chart . 18

3.1 Data packet received from CyberGlove-HT 29

4.1 Number of trials for each class 53

4.2 Number of samples for each class 54

4.3 Number of trials for training and test 58

5.1 Confusion Matrix 66

Chapter 1

Introduction

1.1 Context and Motivation

Immersive technology implies a technology that attempts to emulate a physical world through the means of a digital or simulated world by creating a surrounding sensory feeling, thereby creating a sense of immersion. “Virtual reality (VR) refers to a computer-generated digital environment that can be experienced and interacted with as if it were real.” “Augmented reality (AR) adds cues onto the already existing real world, and ideally, the human mind would not be able to distinguish between computer-generated stimuli and the real world.” “Mixed reality (MR) is the merging of real and virtual worlds to produce new environments and visualizations, where physical and digital objects co-exist and interact in real time [5].”

In today’s world, this technology has been used in many fields such as engineering, science, art, education, training and learning. This technology is being used to overcome the requirements of specific areas and tools for any scientific experiment. On the other hand, it is being used in psychology research to understand human responses and behaviors. Due to its huge potential, it has been a wide and attractive field for many researchers over the last 50 years.

Because interaction tasks enhance the immersion of the user, interactivity with the virtual scenarios has been considered as a very essential point for any Immersive Environment Application. Over time, with the increase of digital media capacity, new approaches regarding interaction tasks have become an essential requirement for immersive technologies. Therefore, new interactive immersive technologies are one of the milestone areas for researchers that can be applied in numerous scientific, educational and art domains. There are a few components that should be carried out in order to enhance the immersive level of a virtual world [6]. One of them is, being in a stable spatial environment with all of a user’s sensorial responses in VR matching with those in the real world. Besides, self-embodiment is another component which is the perception that the user has a body within the virtual world. A lot of researches show that when the users are given a virtual body that properly matches their movements, they quickly realize that level of immersive is enhanced. On the other hand, physical interaction is one of the most important components that make users believe they are in an alternative world where they can not only look around but interact with virtual objects. Nevertheless, further research is required in order to improve self embodiment and interactivity. With the same motivation, in this master project, glove-based immersive interaction tasks in 3D scenarios were developed based on a cyber glove acting as a sensing device of physical movements and its approach will be presented.

1.2 Main Developments and Contributions

This master project work was developed as a part of an R&D project named Human Tracking and Perception in Dynamic Immersive Rooms - HTPDIR[1].

The HTPDIR Project proposes a low-cost system to map static and dynamic obstacles in the physical space to let users be aware of the limitations in the real world while he or she experiences the virtual world. Alongside, the project offers a glove named *CyberGlove-HT*, which allow the user to perform interaction tasks in an Immersive Environment.

In this master project, the main contribution is the development of interaction and manipulation tasks, in 3D immersive scenarios, such as grabbing objects, open/close doors and drawers, based on hand, wrist and finger movements detected with the CyberGlove-HT. The level of naturalness of this approach was compared to the HTC controllers.

Additionally, it was developed and assessed a module to detect horizontal and vertical swipe gestures, based on inertial sensors.

In Figure 1.1, the main modules and functionalities of the overall system (specific modules developed in this work are highlighted in blue is presented as an overview diagram regarding the interaction tasks part). A hand model has been developed for manipulation tasks, according to manipulate according to the data received from *CyberGlove-HT*. By using the physics engine of the 3D real-time development platform,

[1]http://htpdir.com/

(Unity 3D), objects have been made available for users to interact with it. To enhance the level of immersion of users, audio is also used. Moreover, haptic feedback is used to select different interaction tasks.

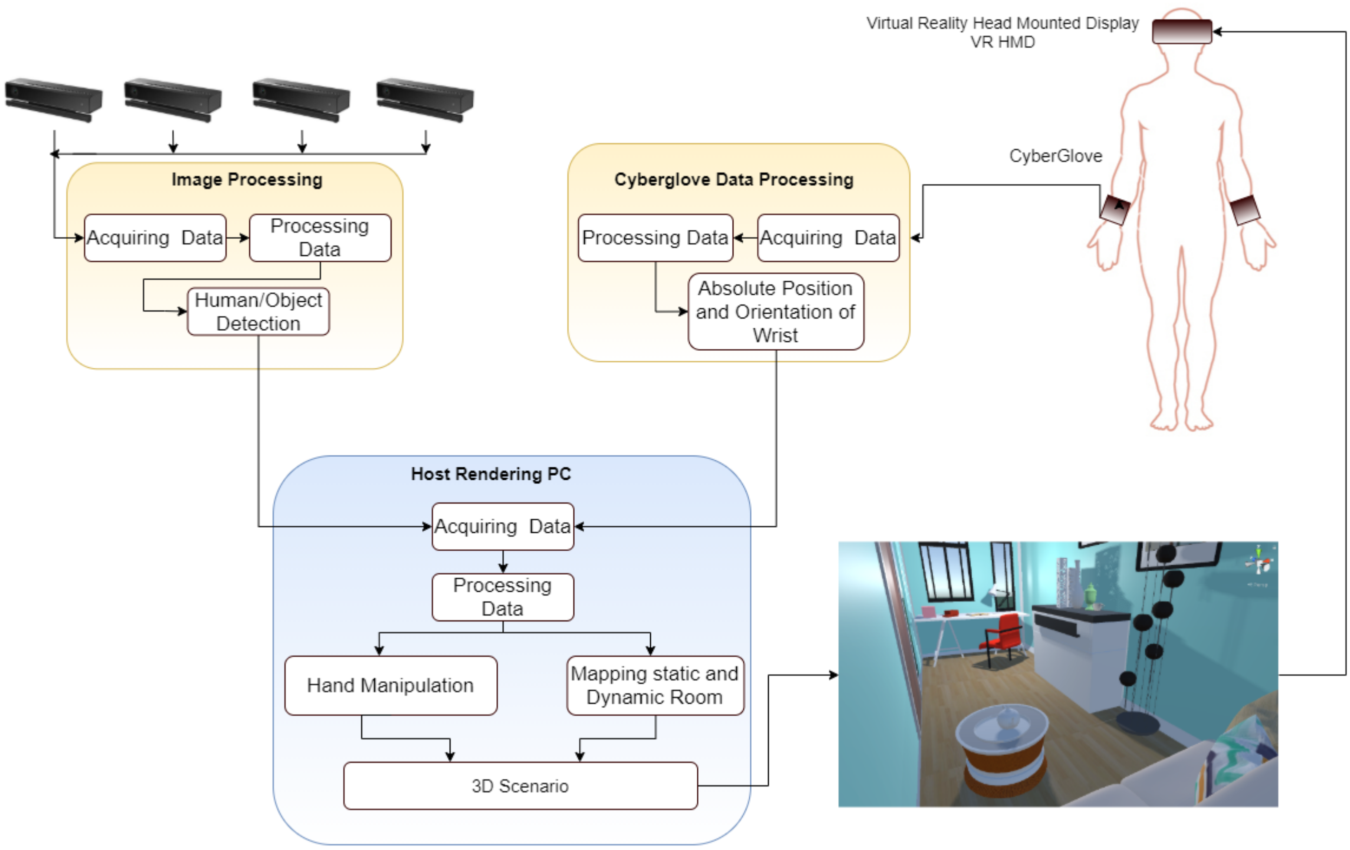


Figure 1.1: Overview diagram of the immersive reality system

1.3 Document Structure

This Master report is divided into six chapters and one appendix as follows:

Chapter Two:

In this Chapter, the State of The Art, literature review, concepts and background related to Immersive Technologies namely, VR, AR, and MR are presented.

Chapter Three:

This chapter describes the design of Interactable 3D Scenarios, the architecture of the system and the integration of each component.

Chapter Four:

In this chapter, the IMU-based Gesture Recognition is presented.

Chapter Five: In this chapter, the experimental results of Interaction and Gesture Recognition are presented.

Chapter Six:

Lastly, chapter 6, draws some conclusions about the work developed, some contributions and future work considerations.

Chapter 2

State of The Art, Background Methods and Materials

Immersive technologies such as Virtual Reality and Augmented Reality have been used widely today in engineering, technology, science, architecture, education and in art fields. It has become one of the important technologies to be discussed regarding its applications, usage, and benefits in the real world. This chapter addresses the background of VR and AR, concepts of presence and immersion, challenges and applications related to the work in the context of interaction with immersive environments.

There are key concepts related to Virtual Reality, which are immersion and presence. Immersion refers to the degree of physical stimulation on the sensory systems and the sensitivity of the system to motor inputs. "The level of immersion is determined by the number and range of sensory and motor channels connected to the virtual environment, and the extent and fidelity of sensory stimulation and responsiveness to motor inputs (ex: commands through the body and head movement, and hand gestures) [7].

A taxonomy has been proposed, by Milgram and Kishino in 1994 [8], that describes the Reality-Virtuality Continuum as shown in Fig. 2.1. According to this continuum, the real environment (RE) stands at one of the ends of this continuum where the scenario is completely our physical world and the virtual environment (VE) stands at the other end where the scenario is totally digitally rendered. Augmented reality stands closer to the VE in this continuum. Mixed Reality (MR) is considered as the reality that takes place between these two ends. It combines the physical and virtual world and provides their components to interact with each other.

The key to defining virtual reality in terms of human experience rather than technological hardware is the concept of *presence* [9]. *Presence* is considered the psychological product of technological immersion [7]. Besides immersion, there are several definitions and theories that have been proposed for the concept of*presence*. *Presence*

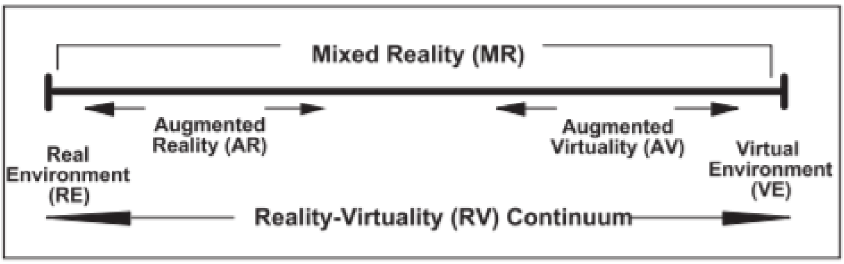


Figure 2.1: Milgram's reality-virtuality continuum *(taken from [2])*

definitions refer to it as the "sense of being there", sense of being in an environment or a virtual environment even when one is physically situated in another [10]. Sheridan [11] describes presence as being in a computer-generated world for presence. Schloerb, [12] distinguishes two types of presence: *subjective presence*, in which the person questions himself to be physically present in the remote or virtual environment and *objective presence*, in which the person can successfully complete a task.

2.1 Concepts and Background Virtual Reality

2.1.1 Defining Virtual Reality

Virtual reality is an immersive computing technology, which provides a unique way to interact with the digital environment and enable people to experience a virtual world immersively. Most popular definitions of virtual reality make reference to a particular technological system that includes a computer capable of real-time animation, controlled by a set of wired gloves and a position tracker, and using a head-mounted device [9].

Virtual Reality is electronic simulations of environments experienced via head-mounted eye goggles and wired clothing enabling the end-user to interact in realistic three-dimensional situations [13].

Virtual Reality is an alternate world filled with computer-generated images that respond to human movements. These simulated environments are usually visited with the aid of an expensive data suit that features stereophonic video goggles and fiber-optic data gloves [14]. .
On the other hand, besides the technological features and requirements of VR, it also

has been defined from the aspect of presence and immersion. *"The VR is an emerging computer interface distinguished by high degrees of immersion, trustworthiness, and interaction. The goal of VR is making the user believe, as much as possible, that he is within the computer-generated environment [15]."*

2.1.2 Historical Background of Virtual Reality

The first time the term of VR was used in 1965, in a paper entitled “The Ultimate Display” published by Ivan Sutherland who described how one day the computer would provide a a window into virtual worlds [16]. Several systems marked the history of virtual reality, as described in the following:

- Sensorama was a machine invented in 1962, by Morton Heiling (See in Fig. 2.2). The Sensorama was the first way to explore the system of VR. The system of Sensorama consisted of multi sensors that could merge a previously recorded chromatic film with smell, sound the wind and related vibration. It had most of the features of such environment, but without interaction [17].

Figure 2.2: Sensorama Simulation Machine

- The Ultimate Display was invented in 1965 by Ivan Sutherland (See in Fig. 2.3. Sutherland suggested a system consisting of interactive graphics, smell, sound and force feedback. He described the Head Mounted Display (HMD) as a window for the VR [18].

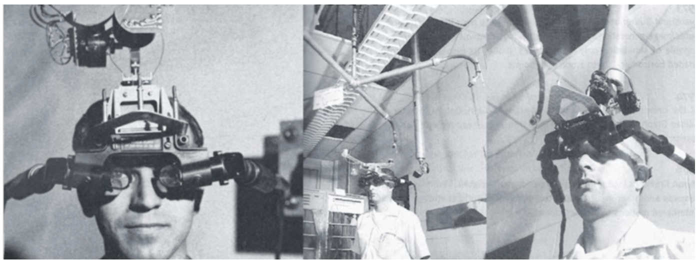


Figure 2.3: The Ultimate Display

- VCASS was a flight simulator which was developed in 1982 by Thomas Furness as the "Visually Coupled Airborne Systems Simulator". The "Super Cockpit" which is the second phase of his project was added higher-resolution graphics and a responsive display [19].
- DataGlove was created in 1985 and the Eyephone HMD created in 1988 by VPL company as the first commercially available hardware of VR for the public [20].
- BOOM was created in 1989 by Fake Space Labs. It stands for Binocular Omni-Orientation Monitor. BOOM was "a small box containing two CRT monitors that can be viewed through the eye holes". In the system of BOOM, the user is able to take the small box with his/her eye movements, move it through virtual environments and keep track of the box by the eye orientation [21].
- Virtual Wind Tunnel was created in 1990 at NASA Ames to allow the monitoring and investigation of flow fields included with BOOM and DataGlove (See in Fig.2.4). Thus, scientists were able to watch and analyze the dynamic behaviour of air flow [22].

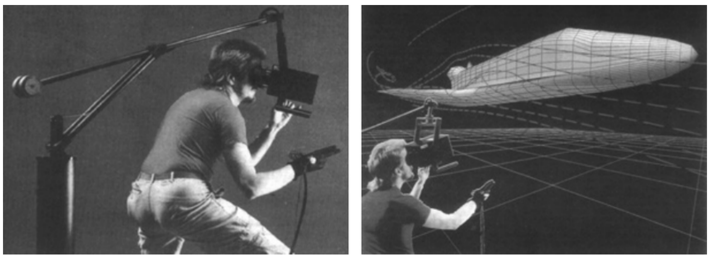


Figure 2.4: The Virtual Wind Tunnel

- CAVE was invented in 1992 as "a VR and scientific visualization system." Instead of using HMD, it uses stereoscopic pictures on the walls of the room. In the CAVE system, the user has to wear LCD shutter glasses [23]. In the CAVE "projection on all six surfaces of a room allows users to turn around and look in all directions.

This allows the user to interact with a virtual environment in ways with a better sense of full immersion" [21].

In this section, a historical overview of Virtual Reality is presented. In Section 2.2, the current technologies such as head mounting devices, tracking systems, controllers and rendering systems are further explained.

2.1.3 Main Trends and Applications: Challenges and Achievements

Thanks to advances in technologies, Virtual Reality is considered as one of the most emerging and appealing ways of interacting with applications. The motivation to create these applications and the growing needs of virtual reality relies on the benefits of simulating the real world dynamically by use of computer software, hardware and virtual world integration technologies. Therefore, the user can have experience of presence by being a part of the action on the virtual safe environment, without any danger. This advantage leads to visualizing a working environment where people actually can not work due to several limitations and conditions.

Virtual reality is present in a wide range of applications of important fields such as business and marketing, medical, education and training, architecture design and prototyping, military applications, mobile and gaming applications, engineering, competitive sports application.

Virtual reality is being used in a number of ways by the business community like virtual tours into a business environment [24]. It is being used in marketing to present products by using a 360-degree view of it. In training, it has been used to allow the trainee to improve their skills without the consequence of failing the operation [25]. These facts provide a better perspective of the design and it helps to reduce the time and cost factor in the engineering and designing process [26].

Education is another area where virtual reality has been adopted for teaching and learning situations. It enables a large number of students to interact with each other as well as virtual objects within a three-dimensional environment. It is able to present complex data in an accessible way to students which is both easy to learn and fun [27]. This type of technology is mostly used in the school of medicine to develop surgery simulations where students can explore more, without stressing the patient. [28].

Furthermore, there are many studies that allow the experience of VR as a stimulus the same way in the real world, with high realism. This is the main reason for VR is widely used in research on applying psychological treatment or training. For instance, or example, treatment of phobias [29].

2.1.4 Challenges in Virtual Reality

Even though VR promises great opportunities, it is still not perfect from a technological, organizational and psychological point of view since it suffers from a number of weaknesses:

- VR requires intensive graphics capabilities. So that, high powerful computer systems and processors are required for creating a proper virtual environment.
- As technologies are growing at a rapid rate, many people are still unaware of such new technologies, along with their advantages. VR is often considered as a game, which is not taken very much seriously. Students can show attitude as it is a game, instead of considering it as an education tool.
- VR is often delivered as propriety solutions that may not be matched with similar environments from other developers. Many companies offer their own tools to create VR environments that are not compatible with the rest regarding hardware and software.

2.1.5 Related work: Interaction Tasks in Immersive Environments

In Immersive environments, such as Virtual Reality (VR), Presence, i.e feeling of "being there" is the essential condition. The user has to be convinced to feel the perception of being in a simulated world. There are several methods in order to enhance Presence in an immersive environment and one of them is to allow the user to interact with the virtual world. In this way, the virtual world is no longer a scenario the user is able to only walk around, he/she is able to interact as well. Interaction techniques can be classified into two categories [30]:

1. Exocentric metaphors, in which users interact with VE from outside. World-In-Miniature and Auto Scaling techniques are examples of exocentric metaphors.

2. Egocentric metaphors, in which users interact with VE from inside and known as the most common technique. There are two methods of interacting in egocentric metaphors:

   - Virtual hand metaphor, which user can grab or touch a virtual object with a virtual hand navigated by one to one mapping between real and virtual hand which is known as "Classic hand technique" or by non-linear mapping as in "Go-Go" technique which allows users to reach further distances in VE.
   - Virtual pointer metaphor, in which the user interacts with virtual objects by pointing at them. A laser ray is used as the pointer and the direction of this ray is determined by the orientation of the hand.

Virtual reality has a lot of applications that benefit from natural interaction techniques. Utilizing hands to manipulate virtual objects efficiently and naturally in virtual environments has an impact on immersion and it is still a challenge for the researchers in this field[31]. In this aspect, there are many studies made to compare and investigate how direct and indirect manipulations are related to the efficiency of performance and realism, naturalism.

Mainly there are several systems to capture the behaviors of the user's hand [32]. One is the optical motion capture system which the video cameras are being used to track the user's hand motions and gestures. Another one is the electromechanical motion capture system which several electromechanical devices, such as an inertial measurement unit (IMU) are utilized to track the gesture and motion of the user's hand [33].

J. Lin and P. Schutze [31], have developed a gesture interface using a Leap Motion finger tracker attached to an Oculus Rift DK2 and implemented three ways of interacting with objects: innate pinching, magnetic force, and a physical button attached to the index finger. Their aim was to compare grasping gestures for direct manipulation, magnetic grasping for remote manipulation, and interacting with objects via buttons in VR. They provided Object color visual feedback to the participants, which was used to balance the error rate by highlighting the object. According to their development [31], one of the limitations that are reported is that The Leap Motion is used for real-time hand tracking, but it can not detect a hand when it is occluded because the controller uses optical sensors and infrared light for tracking. On the other hand, when the leap motion initially detects a hand, it often identifies it as the wrong hand which poses a challenge for the application. In conclusion, the initial tests in a pilot study showed that the participants felt the grasping was more natural, but that the button was more reliable.

In the study of Yang Wenzhen et al. [34], a development of the master-slave hand system is proposed, which includes precision position tracker (PPT) interface, glove interface, dexterous virtual hand, and simplified virtual hand manipulation intentions, and so on. According to this paper, the PPT interface can accurately obtain the position and orientation of the user's hand. The glove interface they proposed can dynamically capture the user's finger joint motions in time. The dexterous virtual hand we modeled is similar to the real hand with 15 finger joints. They used the Oriented Bounding Box (OBB) Collision Detection Algorithm and defined four manipulation intentions as contacting, grasping, moving, and releasing. For interaction, their proposed method tests if the OBB collision detection algorithm has detected that any finger of the dexterous virtual hand touching a virtual object, this means the dexterous virtual hand contacts with the virtual object. Then, if there are two or more fingers touching the virtual object, one of them must be the thumb, the dexterous virtual hand grasps the virtual object. In the grasping case, the virtual hand can move the virtual object in virtual environments. Once the grasping condition is not satisfied, the virtual hand releases the virtual object. Even though their hardware was expensive, the obtained experience results show the master-slave hand system proposed a natural and efficient user-interface for users interacting with virtual environments.

Y. Kim and J. Park present a talon metaphor for bare-hand virtual grasp and release, which does not rely on any wearable devices, sensors, and markers [35]. In this paper, the proposed method uses one RGB-D sensor for both tracking hands and manipulating a virtual object yet, it enhances naturalness and reduces fatigue on users by solving sticking object problems and residing inside of an object problem. Their algorithm calculates optimal grasping and releasing states using finite virtual rays, which provide rapid and accurate selections in comparison to natural interaction metaphors. Finite virtual rays project out from the middle of the tips of the thumb, index, and middle finger. Their method calculates the intersection points of two pairs of virtual rays inside of an object for grasp.

Lin el at. [36] considered how concepts related to the virtual hand illusion, user experience, and task efficiency are influenced by the various size of a user's actual hand (Small, Fit, Large) and the virtual hand. Furthermore, they compared two-level interaction modalities using finger motion and a hand-held controller. By using the

Glove, participants were asked to grasp the virtual blocks and their hand motions are tracked. In the controller condition, they were asked to press the buttons to grasp the blocks. Their study results showed that being able to directly control virtual hands rather than use a controller induces a stronger level of virtual hand ownership and thus increases the virtual hand illusion while the controller provides more precise performance in detection. According to them, one of the reasons participants feel more in control with the controller may be the haptic feedback obtained from the button. They reported that their results do not support any main effect of hand size on the virtual hand illusion or task efficiency.

There are mainly two different types of grasping methods in VR [37]. The first one is to simulate soft tissues of the hand and the material of the virtual object using a physics-based simulation with collision detection. Although this method provides a very realistic manipulation, it requires significant computing resources, making it difficult to run in real-time. The other method is called rule-based grasp. This method requires a set of pre-defined rules, a grasp, or a release is triggered once specific rules are satisfied. Even though it is straightforward to compute, it provides very limited user experiences in terms of realism. Liu el at. [37], presents a design of a glove based manipulation system that combines these two methods with reasonable balance. According to the geometry of the collision between the virtual hand and virtual objects, a caging-based approach is integrated to determine a stable grasp. The experiments were resulting in a significantly higher success rate in grasping and moving objects in VR, compared to the popular LeapMotion sensor. This experiment also indicates that the proposed design is more robust in terms of the grasp types: power grip, cylindrical grasp and extension grip. Although the LeapMotion sensor performs well for Power Grip with thumb abducted type of grasp, it is limited in the other two types of grasps. In contrast, the grasp using the proposed design performs well for all the grasp types.

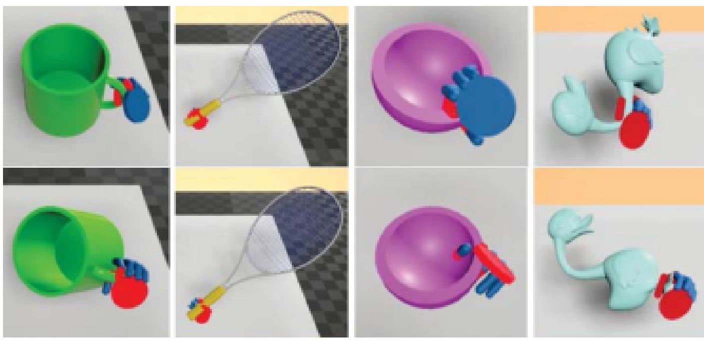


Figure 2.5: Different type of grasps: Grasping a mug and a toy with power grip, a racket with cylindrical grasp and a bowl with an extension grip.

One of the fundamental problems of hand-based interaction is Hand-object interpenetration, where a hand sinks into virtual objects due to the lack of real physical constraints [38]. In this paper, Prachyabrued and Borst described the visual feedback techniques and summarize the results of a pilot study where they compare techniques to OH (outer hand) and IH (inner hand). Figure 2.6 illustrates to identify which technique is most consistently found better than baselines, or at least provides the most promising tradeoff. The pilot study exhibits that, allowing visual interpenetration produces a lighter touch than a visually-constrained virtual hand (outer hand, OH). Even though, OH is a standard approach to visually mimic real-world grasping, it appears to be the worst performer. While IH performs the best but is subjectively bad. The goal of additional visual feedback is to balance this tradeoff.

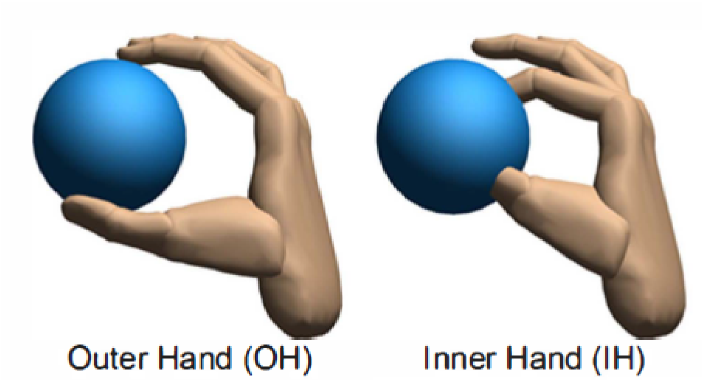


Figure 2.6: Hand-object interpenetration.

Moehring and Froehlich [32], have focused on the comparison of finger-based direct interaction to controller-based ray interaction in a CAVE as well as in head-mounted displays (HMD) in respect to the relative performance and the advantages and disadvantages of both approaches. They provided tactile and vibrotactile feedback to enhance robustness and feedback. The results of their study show that finger-based interaction is preferred over indirect interaction for the assessments of various functionalities in a car interior. While controller-based interaction is more robust, it lacks realism.

Ullmann el at.[39] presented a real-time approach for grasping rigid virtual objects in an intuitive manner. They enabled the possibility of changing between one-hand and two-hand grips, according to the natural grasping behavior of humans. The grasping conditions are required for the different grasping decisions, namely, condition 1 handles the fundamental requirements and condition 2 the duration of the grip.

2.2 Technologies: Head Mounting Devices, Tracking Systems, Controllers and Rendering Systems

In this section, current technologies in VR including the hardware systems and rendering systems are described.

2.2.1 Hardware Systems

Virtual Reality Hardware technologies are addressed in two main categories as input and output devices.

Output Devices: The main category in output devices is Visual Display technology. These displays are commonly known as Head Mounted Device (HMD). There are mainly three subcategories in HDMs which are mobile HMD, wired HMD and standalone HMD. The mobile HMDs provide a simple casing by keeping the smartphone at a determined distance from the lens and displaying 360-degree panoramas rendered from a stable point of view. Also alternatively they might provide interactive walk-throughs based on gaze-directed navigation. While Google Cardboard which is one of the first developed HDM, works as a basic viewer, Samsung has developed a GearVR smartphone holder which provides an additional touchpad on the side of the case in collaboration with Oculus. (See in Fig. 2.7) On the other hand, Samsung has upgraded the resolution and image quality with Odyssey.[1]

Figure 2.7: Google Cardboad is on the left and GearVR is on the right[1]

Furthermore, Wired HMDs and Standalone HDMs are all equipped with additional sensors such as accelerometers, magnetometers and gyroscopes in order to obtain accurate optical tracking by using sensor fusion to combine this information. There

[1]https://www.maxboxvr.com/magazine/google-cardboard-vs-samsung-gearvr-vs-samsung-odyssey-virtual-reality-headsets

are mainly three big competitors currently: Oculus Rift[2], PlayStation[3], and the HTC Vive[4]. Wired HDMs are physically connected to PCs such as the HTC Vive Cosmos, PlayStation VR, and Valve Index, while standalone HDMs offer higher physical freedom by completely removing the cables and not requiring any external device to handle processing such as The Oculus Quest 2. The Oculus Rift is one of the biggest names in the VR market and Oculus Rift S is their most upgraded wired HDM product. Lately, the company focused on the Oculus Quest 2, a standalone headset with an option to be linked to a PC. Even though the processing is not as powerful as a gaming computer, the Oculus Quest 2 still provides smooth graphics. It offers a comprehensive VR experience with no wires needed. and currently provides a resolution at 1,920 by 1,832 per eye. It has two motion controls for full 6DOF head and hand motion tracking. (See in Fig. 2.8). The HTC Vive headset contains two AMOLED screens that have 1080x1200 resolution streaming data at high frequency of 90Hz and a field of view of 110 degrees to create the sense of 3D virtual reality.

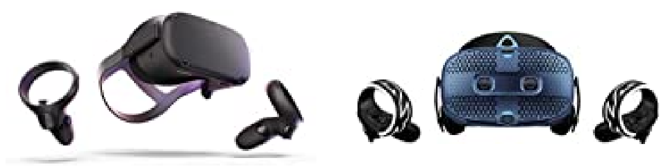


Figure 2.8: Oculus Quest VR is on the left and HTC VIVE Cosmos is on the right

HTC Vive HTC Vive provides similar specifications to the Oculus Rift, their main difference is in the tracking range. Opposed to many other wired HDMs, HTC Vive provides a room a 5m x 5m space to be tracked and allows users walking within the virtual environment. HTC's Vive Cosmos is the upgraded version of the Vive headset, boasting a higher resolution and replacing the external base stations with outward-facing cameras for motion tracking. The table 2.1 compares the latest products of these two big companies.

Input Devices: Input Devices provide different and diverse features like output devices, for instance, controllers. VR controllers supply an interactive experience via buttons, triggers, and tracking systems to users as traditional controllers. Furthermore, they have new approaches. For example, Oculus Touch provides a basic gesture and finger movement recognition and high precision which enables the user to have a true

[2]https://www.oculus.com/rift/
[3]https://www.playstation.com/en-us/ps-vr/
[4]https://www.vive.com/us/

Table 2.1: HTC VIVE Cosmos vs Oculus Quest VR Headset Comparison Chart

Features	***HTC VIVE Cosmos***	***Oculus Quest VR***
Headset Type	PC-tethered	Self-contained
Display	Dual LCD	Dual LCD
Resolution	2880 x 1700	3200 x 1400
Field of View	110°	110°
Refresh Rate	Up to 90 Hz	Up to 72 Hz
Tracking	6 tracking cameras, 6DOF	4 tracking cameras, 6DOF

hand precise experience. The HTC Vive controller uses touchpads which provides high precision and is tracked with the Lighthouse system (See in Fig. 2.9).

Figure 2.9: Oculus Touch controller is on the left and HTC VIVE controller is on the right.

Tracking Systems are another component of input devices. Although there are several tracking methodologies, the main trend is optical tracking. Optical tracking can be done either with or without markers. Tracking with markers involves several reference points, and cameras constantly detect these markers and then use various algorithms to extract the position and orientation of the object. Markerless tracking uses the natural features of the environment to determine position and orientation. Currently, there are two methods that are used in the market. Oculus Rift uses external stationary cameras that track the position and orientation of the headset and controllers by detecting the IR LEDs on the devices. This method is accurate and has low latency. Oppositely, the HTC Vive headset has a camera inside. The lighthouses emit laser arrays that sweep the area horizontally and vertically and sensors on the headset and controllers can detect these sweeps and use the timings to determine position [40]. Although this method allows larger player areas, due to the higher processing requirements, the latency can be higher. Oculus Quest employs markerless tracking by creating a 3D map of the environment in real-time by using the SLAM (Simultaneous

Localization and Mapping) process.

2.3 Virtual Reality Software Frameworks

Hardware devices are used together with several different open platforms for software development. Oculus provides its software development kit (SDK) on GitHub. Likewise, OpenVR is another software development kit provided by Valve. The SteamVR[5] platform uses it as an interface API between hardware and software. OpenVR SDK[6] and SteamVR SDK[7] are available under GitHub.

A virtual reality game engine, allows game developers to design, build, and test their games by using several virtual reality SDKs. These tools enable developers to create and edit 3D characters, objects and fully immersive 3D experiences. The two most popular game engines so far are Unity[8] and Unreal Engine[9]. Both of them enable design and edit interactively immersive environments. Nevertheless, the choice of the developer depends on the requirements. Unity has access to a wider variety of plugins and asset store. On the other hand, Unreal provides better graphics quality. The language of Unity is C# which is a higher-level language than C++ used by Unreal, and the overall engine architecture is simpler in Unity, which facilitates and speeds up the process of development.

Unity 3D is a game engine that can be used to create three-dimensional, two-dimensional, simulations as well as immersive environment applications such as virtual reality, and augmented reality games.[10]. Unity 3D is used for scripting, scene creation, animation, app architecture development, level design, motion design, and physics implementation. The application developed in this project used Unity 3D, the version 2017.3.1f1 Professional.

[5]https://store.steampowered.com
[6]https://github.com/ValveSoftware/openvr
[7]https://github.com/ValveSoftware/steamvr_unity _plugin
[8]https://unity.com/
[9]https://www.unrealengine.com/en-US/
[10]https://en.wikipedia.org/wiki/Unity 3D_(game_engine)

2.4 Classification Methods for Gesture Recognition

Classification is one of the most common decision-making tasks of human activity which is used to assign an object into a predefined group or class based on a number of observed attributes related to that object [41]. There are mainly two categories in classification methods, sequential and non-sequential classification methods. Non-sequential methods are characterized by individual cases in each training or test dataset. Bayesian Model, K - nearest neighbor, Regression Tree, Support Vector Machine (SVM), Artificial Neural Networks (ANN) are the commonly used non-sequential classification algorithms. Even though these algorithms are originally made for nonsequential data, they can be fed by a feature vector with a time window corresponding to a time series. Besides, sequential methods take into consideration the time factor. It is possible to model time, perform motion analysis and recognize human activity with sequential methods. Markov Models, Recurrent Artificial Neural Networks (RNN) and Long Short Term Memory network (LSTM) are sequential algorithms.

LSTM was proposed by Hochreiter and Schmidhuber [42] as a special RNN type which is stable and capable modeling long-range dependencies [43]. On the contrary to RNN, LSTM prevents rapid gradient vanishment which is a challenge for RNN. The architecture is composed of a forget gate, input gate and output gate. Forget gate is responsible for throwing away information that is no longer needed from the cell state. The input gate is responsible for adding information to the cell state. The output gate selects useful information as an output from the current cell state. The memory cell ct accumulates the state information. The self-parameterized controlling gates make the decision whether the information will be forgotten or pass to the next cell. If the input gate (i_t) is activated, the information will be kept in the cell. The decision of throwing away the information from previous cell memory (C_{t-1}) is made by forget gate (f_t) depending on previous cell output (h_{t-1}) and input vector (x_t) (see Fig. 2.10).

The output gate controls whether the current cell state information is visible. The formalization of the input gate, forgate gate and output gate layers are presented in Eq.2.4, Eq.2.4 and Eq.2.4 where W_i, W_f and W_o are weight parameters and b_i, b_f and b_o are bias parameters.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.1}$$

[11] `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`

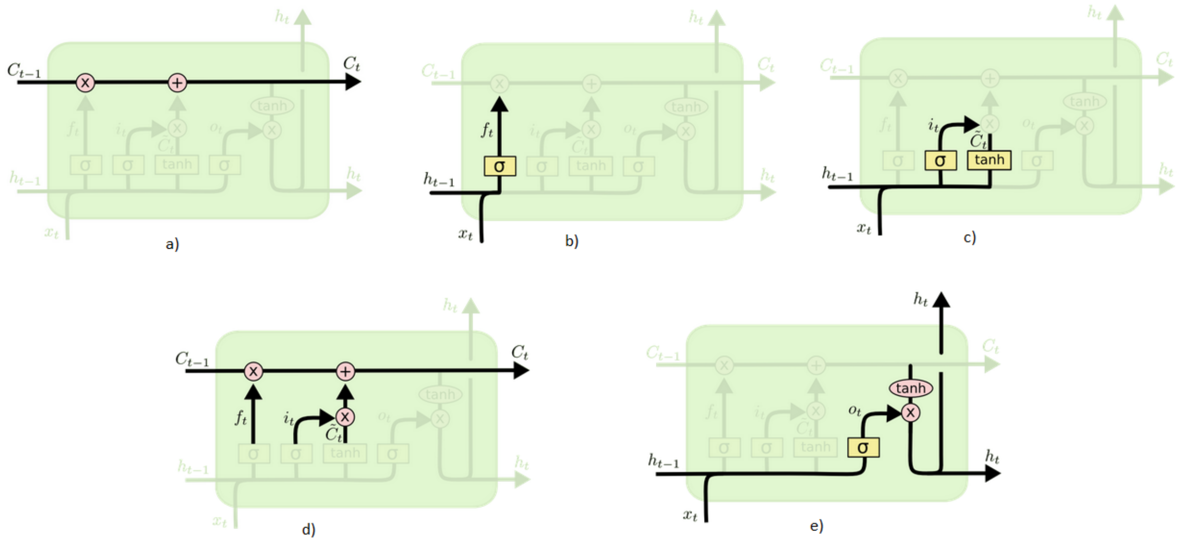


Figure 2.10: LSTM Network Workflow[11]: a) the cell state; b) the forget gate layer; c) the input gate layer d) update the old cell state; e) output of relevant information

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.2}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.3}$$

Chapter 3

VR MODELS, INTERACTION AND MANIPULATION TASKS

In this chapter, the architecture of hardware and software systems and their components are presented. Section 3.1 explains the hardware system, the features of the user tracking system and CyberGlove-HT. Section 3.2 presents communication and data parsing between CyberGlove-HT and Unity 3D and between Kinect Server and Unity 3D. Besides, integration and design of 3D scenarios and interactive objects and different methods of physically-based hand manipulation techniques are demonstrated. Furthermore, this section describes the methods for rendering objects and human bounding boxes dynamically in order to visualize the objects and human tracking.

3.1 Hardware System

3.1.1 Tracking system HTC Vive

The system setup was installed in an IPT room in order to carry out the tests of the development process with an immersive environment. The room was infrastructured and equipped with:

- 1 set of HTC Vive virtual reality glasses capable of providing the user with a 360º view of the virtual scenario consistent with the user's movements and respective controllers.
- 1 wireless transmission system (video, sensors, controllers) for HTC glasses Vive, TPCAST Wireless Adapter, which allows greater freedom of movement in the considered space, free of cables and as such one allows a more natural interaction.

- 4 Kinect sensors (Kinect v2 for Xbox One) capable of acquiring information three-dimensional scenery (static and dynamic scenes)
- 1 computer with high processing capacity and graphics performance capable to control the various sensors in the room, map the 3D environment and synthesize the images to display on the virtual reality glasses display.
- 2 lighthouse base stations which power the presence and immersion of room-scale virtual reality by helping the Vive headset and controllers track their exact locations.

In Fig. 3.1 and 3.2 the Immersive Room and HTC Vive components are shown.

The headset's position and orientation are tracked with a pair of laser-emitting base stations that are called Lighthouses. Two base stations are placed parallel to each other. Hand movements and gestures were tracked either through HTC Vive controllers, or through a wrist-tracker strapped around the wrist.

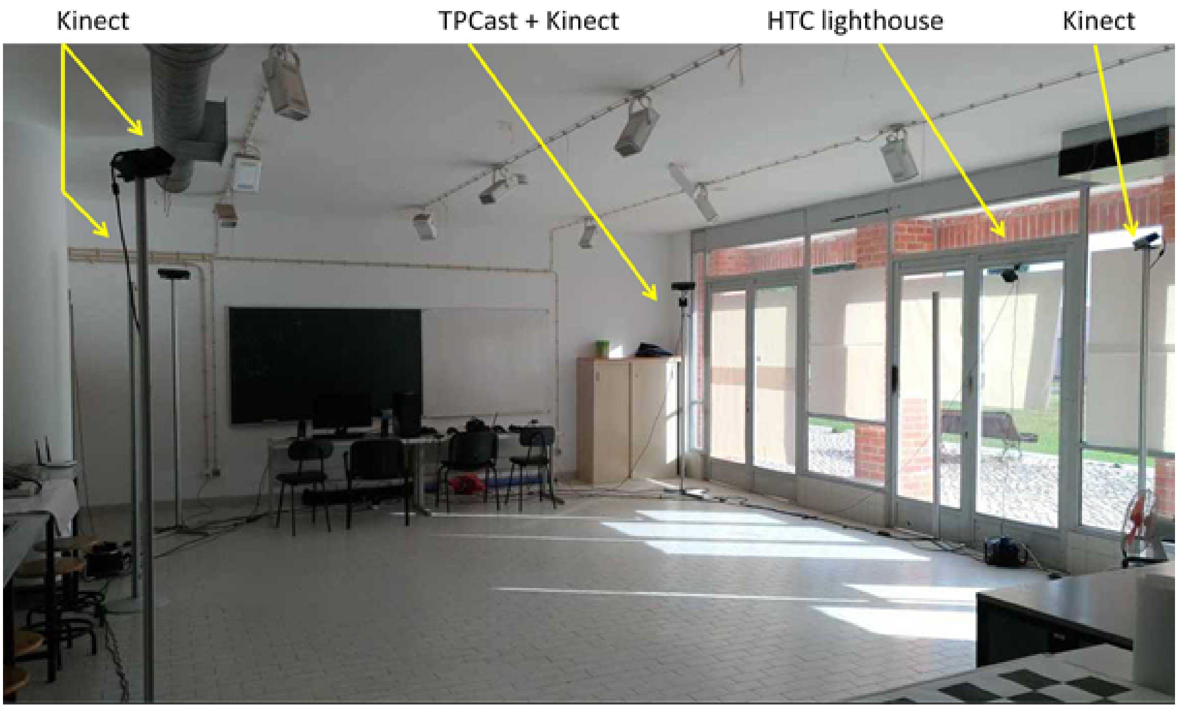


Figure 3.1: The Immersive Room Setup

3.1.2 CyberGlove-HT

The project involves a CyberGlove-HT electromechanical device that allows capturing hand motion such as, movements of the wrists, hands and fingers. The glove system

Figure 3.2: The HTC Vive components and TPCAST Wireless Adapter

can detect the curvature of the fingers without discriminating its joints. It is considered as the main interaction component besides the controllers, by increasing the capacity of natural interaction and on the other hand, they increase the feeling of embodiment.

In Figure 3.3, the components of the device are:

- 1 Sensor IMU (BNO055) with the ARM Cortex M0, detects absolute orientation around x, y, z;
- 1 ESP8266-07 Wi-Fi module with built-in Cadence Tensilica L106 microcontroller allows 802.11 Wi-Fi technology with UDP / IP communication;
- 5 Flexible sensors Flex sensor 2.2, detect finger curve angle;
- 3 Force sensors (FSR 400), to obtain pressure detection in mixed reality scenarios
- 1 micro-laser range sensor (VL53L0X), measures of distance to obstacles;
- 1 Vibro-motor to provide whenever an obstacle is detected below a threshold (safety) or to provide feedback in situations of interaction with virtual objects;
- 1 PPG sensor (MAX30105), Heartbeat detection (possibility to assess emotional reactions) - Not used;
- 1 LiPo BAT525 battery, 3.7 V, 1050mAh;
- 1 HTC Tracker to obtain Absolute wrist position;

The *CyberGlove-HT* provides as output the following data:

- Absolute hand orientation in quaternions - w, x, y, z and acceleration -x, y, z: Information of hand/wrist motion,
- Curvature degree from - 0 to 180º for each finger in the x-axis: Information of if the user closes or opens his/her fingers,
- Finger pressure (touch - ON / OFF states): Information of if the user touches or grabs any object in the real world,
- Distance objects (in mm): Information of distance between the user's hand and any object or obstacle around,
- Heartbeat (bpm): Information of the heartbeat of the user.

Moreover, it gathers an input data of Haptic feedback from *Unity 3D* in order to trigger its Vibro motor.

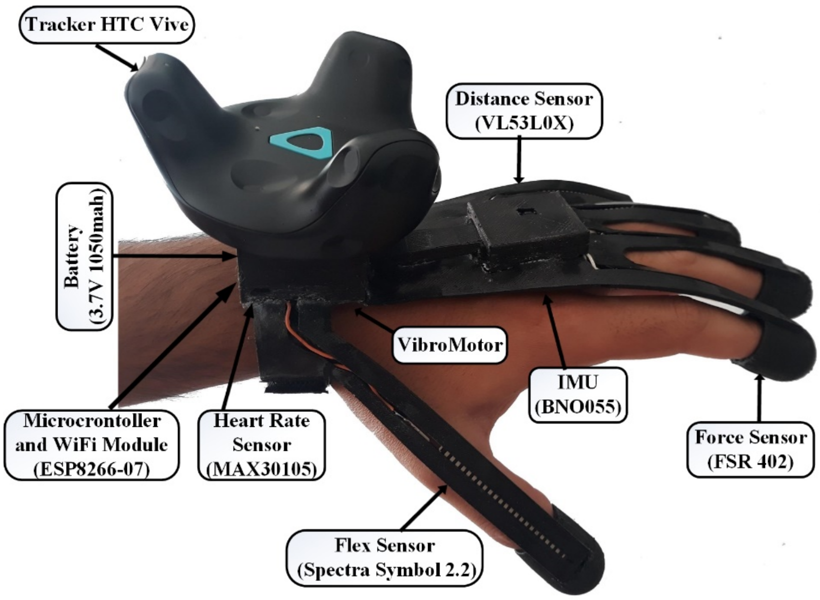


Figure 3.3: The components of CyberGlove-HT

3.2 Immersive Software Architecture

This section describes the Software Architecture of the system(see Fig. 3.4). The pose information of the user's head and user's hand is provided by two different sources, HTC Vive Tracking System and CyberGlove-HT. SteamVR interferences between the hardware and software system in order to provide localization information of the head-mounted device, controllers, and trackers. SteamVR PlugIn is an asset that enables to access these data from Unity scripting, including pose data of trackers which are used to visualize the motion of the user's wrist. On the other hand, CyberGlove-HT provides the absolute orientation of the hand and the curvature degree of the fingers. These data packets are received by UDP communication between Unity and CyberGlove-HT server and parsed with CyberGlove Receiver Script. The data of the absolute orientation of the IMU is used to visualize the motion of the hand. Likewise, the curvature degree of fingers is employed to visualize the movement of the fingers as opening and closing.

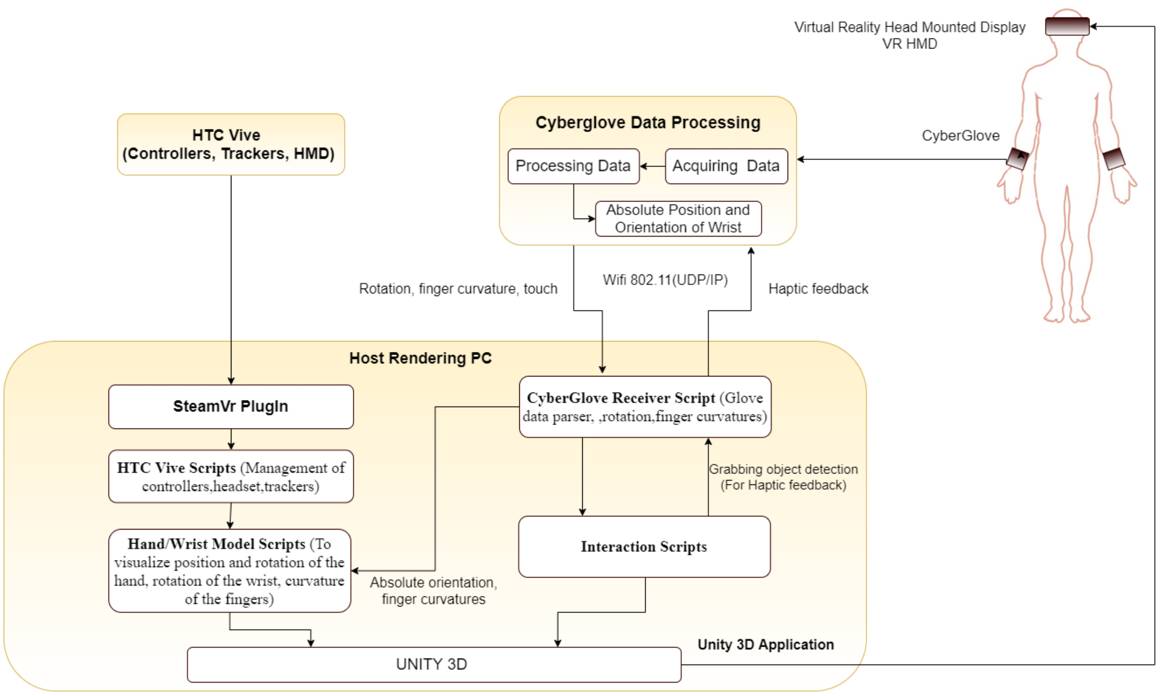


Figure 3.4: Detailed software diagram architecture and its interaction with the sensing devices.

Furthermore, these data are used in order to provide interaction via several scripts. As it is mentioned in section 3.2.2, the *CyberGlove-HT* Server provides information of hand orientation, namely roll, yaw, pitch (α, β, θ), angular acceleration, measurements of the obstacle distance sensor, heart rate, finger curvature, and information pressure sensors located at the user's fingertips. These data are obtained with CyberGlove Receiver script which provides the communication with the *CyberGlove-HT* and it is responsible for forwarding the information of the hand and fingers motion to sub-scripts which are associated with the objects to be manipulated and responsible for providing haptic/visual feedback. (see in Fig. 3.5) .

In Unity3D each thread executes a sequence of programming instructions. The Main Thread runs at the start of the game by default and it can create many new threads to carry out tasks. These new threads run in parallel and synchronize their result with the main thread once it is completed.[1]

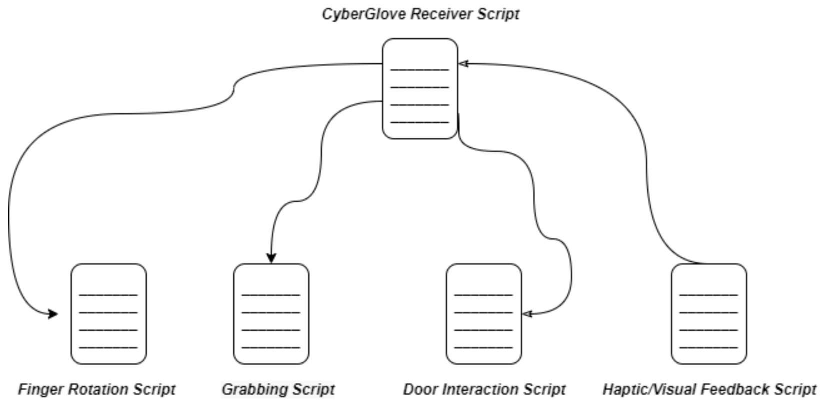


Figure 3.5: Management/interaction of scripts responsible for implementing the actions.

3.2.1 Software Framework

This section specifies the software used in the project are specified. As will be explained in more detail in section 3.2.3 and section 3.2.4, some 3D models have been created using the Unity 3D object generator, and all 3D models have been interactive using the utility of Unity 3D's physics engine. By employing this physics engine, Unity 3D helps to simulate physics to ensure that the objects correctly accelerate and respond to collisions, gravity, and various other forces.

To expand this simulation application into an immersive environment, a virtual reality interference tool called *SteamVR* (version: 1.9.16) is used to be able to use the hardware system. This VR software system supports several headsets including HTC-Vive. In Fig. 3.6, the SteamVr Status Window is shown. This window indicates the status of each component of the hardware, namely, headset, controllers and the lighthouses. The green color means good connection while the grey color means the devices are not connected.

SteamVR provides a room setup tool to determine the area of the immersive environment which allows users to select the limitations of the area. In case of the user gets very close to these limits he/she will see a transparent squared wall as it is shown in Fig. 3.7.

[1]https://docs.unity3d.com/Manual/JobSystemMultithreading.html

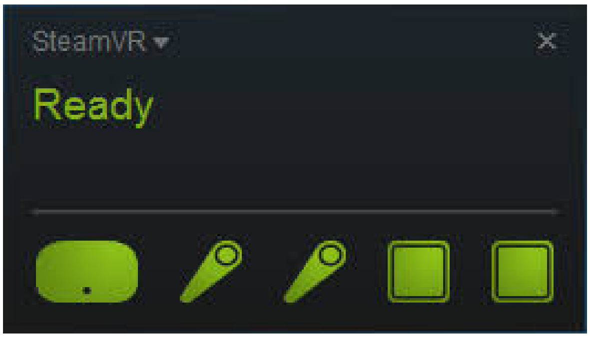


Figure 3.6: The SteamVR Status Window

Figure 3.7: Setup of Immersive Room

A third-party package called SteamVR PlugIn is used as a wrapper to interconnect the hardware system and Unity 3D. It provides access to sensor information provided by the headset and tracking system from the scripts created by Unity 3D.

3.2.2 Communication and Data Parsing with CyberGlove-HT

This project includes two main communication and data parsing processes. This section describes the communication and data parsing between the CyberGlove-HT Server and Unity 3D, while section 3.2.5 explains the communication process between the Kinect and the Unity 3D. This section focuses on how to receive and send data from Unity 3D to the CyberGlove-HT.

Communication between the CyberGlove-HT and Unity 3D, the CyberGlove-HT acts as a server by sending the data packets to Unity 3D. This communication requires a high speed. Due to this requirement, it employs UDP Communication Protocol which decreases the communication overhead. This communication is bi-directional. Unity 3D sends the information whether the Virtual Hand collides with any virtual object in the immersive environment for haptic feedback (See Fig. 3.4), while CyberGlove-HT Server provides information of absolute hand orientation in quaternions, acceleration, curvature degrees of fingers, finger pressure, distance with objects, and heartbeat to Unity 3D in the data packet format shown in Table 3.1.

Table 3.1: Data packet received from CyberGlove-HT

packet																	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Hand	Quaternion				Acceleration			BPM	Distance	Curvature Degree of Fingers					Force of Sensors		
L/H	w	x	y	z	x	y	z	bpm	mm	Finger 1	Finger 2	Finger 3	Finger 4	Finger 5	Finger 2	Finger 3	Finger 4

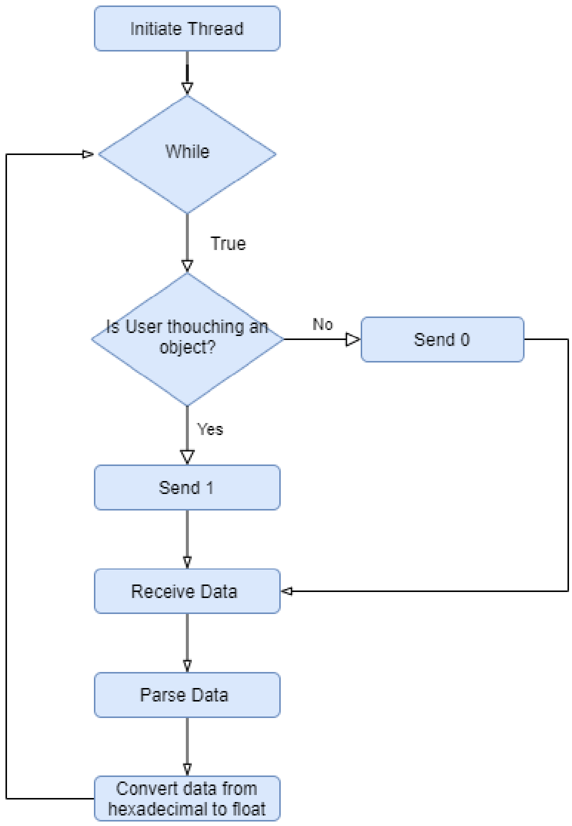


Figure 3.8: The FlowChart of Communication and Data Parsing

The communication and data parsing process flowchart is shown in Fig. 3.8. In each frame, the condition of whether the user is touching any object or not is checked. If he/she is not, then Unity sends "0" and then receives a data packet. However, if the user is touching any object, Unity sends "1". In the last case, the CyberGlove-HT microcontroller vibrates the tactile motors in order to provide haptic feedback. After the data parsing process, the data is converted to float variables. This process of receiving and sending data takes place every 20 ms in a separate thread by using Multithreading programming which takes advantage of a CPU's capability to process many threads at the same time simultaneously. One thread runs at the start of a program by default. This is the "main thread". The main thread creates new threads to handle tasks. These new threads run in parallel and usually synchronize their results with the main thread once completed. Listing 3.1 exemplifies the code to create a thread and how to use thread in the background to prevent application crushes. *ReceiveData* is the function where UDP communication.

3.1: Creating thread

```
//Building Server thread
       receiveThread = new Thread(new ThreadStart(ReceiveData));
//Execute Thread in Background
       receiveThread.IsBackground = true;
//Start the Thread
       receiveThread.Start();
// Build the UDP Client on the port [port]
       client = new UdpClient(port);
       ...
//.NET representation of an IP + Port
       IPEndPoint anyIP = new IPEndPoint(IPAddress.Parse(IP), port);
```

After UDP communication is set up, Unity 3D sends to CyberGlove-HT information to be used as haptic feedback for the user. As in the following part of code, the Unity 3D sends data *"1"* whether the user touches or collides with any virtual object or *"0"* if she/he does not. In Listing. 3.2, *istouch* is a boolean value that controls the collision mentioned.

3.2: Sending haptic feedback information to CyberGlove-HT

```
 if (!istouch)
   {
    //If user is not colliding with any object send 0
    int test = client.Send(Encoding.ASCII.GetBytes("0"),
       Encoding.ASCII.GetBytes("0").Length, anyIP);
   }
  if (istouch)
   {
     //If user is colliding with an object send 1
    int test = client.Send(Encoding.ASCII.GetBytes("1"),
       Encoding.ASCII.GetBytes("1").Length, anyIP);
    }
```

After sending haptic feedback data to *CyberGlove-HT*, Unity 3D receives the data packet provided by *CyberGlove-HT* as shown in Listing 3.3.

3.3: Receiving and parsing the data packets

```
//Convert the data to string
   byte[] data = client.Receive(ref anyIP);
   string text = Encoding.UTF8.GetString(data);
   String[] arrayHex = text.Split(',');
//Converter hexadecimals
    ConvertHexCoord(arrayHex);
```

Afterward, all the received values are converted to float variable as a part of the function presented in Listing. 3.4.

3.4: Data is converted to float variables

```
byte[] inData = new byte[4];
if (inString.Length == 8)
  {
    inData[0] = (byte)Convert.ToInt16(inString.Substring(0, 2), 16);
    inData[1] = (byte)Convert.ToInt16(inString.Substring(2, 2), 16);
    inData[2] = (byte)Convert.ToInt16(inString.Substring(4, 2), 16);
    inData[3] = (byte)Convert.ToInt16(inString.Substring(6, 2), 16);
  }
int intbits = (inData[3] << 24) | ((inData[2] & 0xff) << 16) | ((inData[1]
    & 0xff) << 8) | (inData[0] & 0xff);
byte[] aux = BitConverter.GetBytes(intbits); float aux2 =
    BitConverter.ToSingle(aux, 0);
return aux2;
```

Section 3.2.4 will further describe, how to use these received values to manipulate the virtual hand and interact with the virtual objects in immersive environments.

3.2.3 Integration of 3D Models

This section describes the development and integration of 3D models in the interactive scenario. Taking into account the potential of this application in real estate promotion, models to represent user body parts and virtual scenarios of a house were developed. In this house the user can interact with the VR environment, actually walking on it, going through doors and handling pieces of furniture. For example, opening a drawer and looking at its contents or simply grabbing objects and changing their position in 3D space (further detailed in section 3.2.4).

Some of the 3D models are designed in Unity 3D using its basic 3D object creators, however, Unity 3D is very limited regarding the creation of complex 3D models. Due to this fact, several 3D models are integrated from different sources, available from platforms that publish, share, discover, buy and sell 3D models.

The main and the most important 3D model of the application is the *Hand*[2] (see in Fig. 3.9).It is a "rigged" model which means it relies on a skeleton with joins, so it can move. Joint's values should be continuously provided to animate such models.

Figure 3.9: The Hand Model

The other retrieved 3D model is the *Saloon Model*[3](see Fig. 3.10 and Fig. 3.11) and *The Piano 3D* base model.[4] The keys are created in Unity 3D by us(see Fig. 3.12).

The Drawer (see Fig. 3.14) and *The Door* models (See Fig. 3.13) are created by Unity 3D and the knob model of the door is obtained from a source.[5]. In the next section, will be explained how these 3D models are made interactive.

[2]https://sketchfab.com/3d-models/13-rigged-hand-fbx-dbf0ec6cf6014788b2d6583edb329c58
[3]https://sketchfab.com/3d-models/living-room-2-isometric-lowpoly-ea7928d3f90f4da89e83daf8185ef2c0
[4]https://www.turbosquid.com/3d-models/free-max-model-grand-piano/444507
[5]https://opengameart.org/content/wooden-squared-door

Figure 3.10: The house from behind perspective

Figure 3.11: The house from front perspective

Figure 3.12: The Piano Model

Figure 3.13: The Door Model

Figure 3.14: The Drawer Model

3.2.4 Interaction Tasks

This section describes how to manipulate the Hand model by using the data provided by the *CyberGlove-HT*. Afterward, it explains how to obtain interaction tasks listed below in the immersive room:

- Grab/release an object, rotate it while holding,
- Open and close a door, turn the knob with your fingertips, hold the handle, manipulate the door by pushing or pulling the handle,
- Open (pull) and close (push) drawers,
- Play the piano with all your fingers,
- Interact with menus,
- Provide tactile feedback when the grab mode is active,
- Provide visual feedback when the grab mode is active.

A demonstration video[6] is available to show these interactions in the developed immersive environment.

The hand-animated model has 6 DoF, it is represented through quaternions and its 3D position is provided by an *HTC Vive Tracker*. In addition, the quaternions provided by *CyberGlove-HT* are used to animate the wrist's 3 DoF, namely, roll, yaw, pitch. Both reference systems are interconnected through a translation along the z-axis as it is shown in Fig. 3.15. This articulated model allows representing a more flexible wrist and hand, natural, closer to reality, thus contributing to the embodiment.

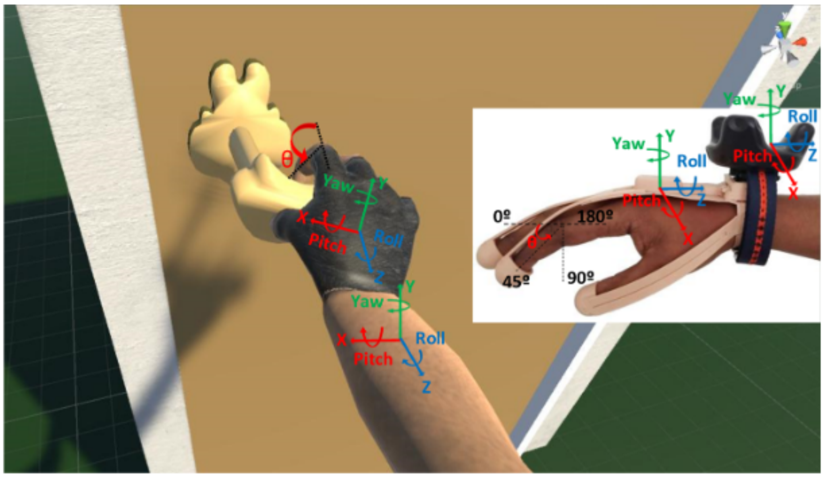


Figure 3.15: Systems coordinates of the hand and wrist in the virtual environment according to an egocentric view

Each finger rotation is animated in relation to the X-axis using the data provided by the flexible sensors. The script presented in Listing 3.5 illustrates the use of use the curvature angle data to rotate the middle joint of fingers smoothly (see in Fig. 3.16).

3.5: Creating finger rotation movement

```
//The finger curvature angle is applied to the middle joint of the finger
Vector3 target = new Vector3(angle , 0, 0);
this.transform.localEulerAngles =
    Vector3.Slerp(this.transform.localEulerAngles, target, smooth *
    Time.deltaTime);
```

[6]https://youtu.be/VIYlYvpz-MM

Figure 3.16: The finger rotation is animated via flexible sensor data

- **Grabbing and Releasing an object:**

The *Grabbing Script* controls the action of manipulating a 3D object with the hand model while wearing the *CyberGlove-HT*. This manipulation involves grasping, rotating and releasing actions. As it is possible to observe from Fig. 3.5, *Grabbing Script* uses the data provided by the *CyberGlove-HT Receiver Script* and determine collision events. Models including hand and graspable objects were made rigid to ensure collision between them in the virtual world. This is achieved by adding Unity 3D's *RigidBody* and *Collider* components, bringing the Unity 3D's physics engine into the play.

Figure 3.17 shows the flowchart of grabbing and releasing interaction. This interaction takes place in an Update function. In each frame, it is being checked whether the user has already held any object or not. If the user is holding an object and if the fingers are closed, the object transform is being set to the user's hand. In case the user open the fingers, the object is released and falls naturally under gravity. I the user's hand collides with an object and closes the fingers without holding any other object, then the object is held until the fingers are opened.

The collision method involve a transparent sphere is created on the hand model in order to check the intersections within determined radius. Listing 3.6 illustrates the script code that validates the grabbing function after contact. *Physics.OverlapSphere* function returns an array with all colliders touching or inside the sphere. Afterward, the finger degree condition is checked. If the angle degree of the index finger is bigger than 35 degrees, the object is accepted as held.

3.6: Detection of the objects close to the virtual hand

```
Collider[] colliders = Physics.OverlapSphere(transform.position,
    GrabDistance);
if (colliders.Length > 0 && angle >= 35)
```

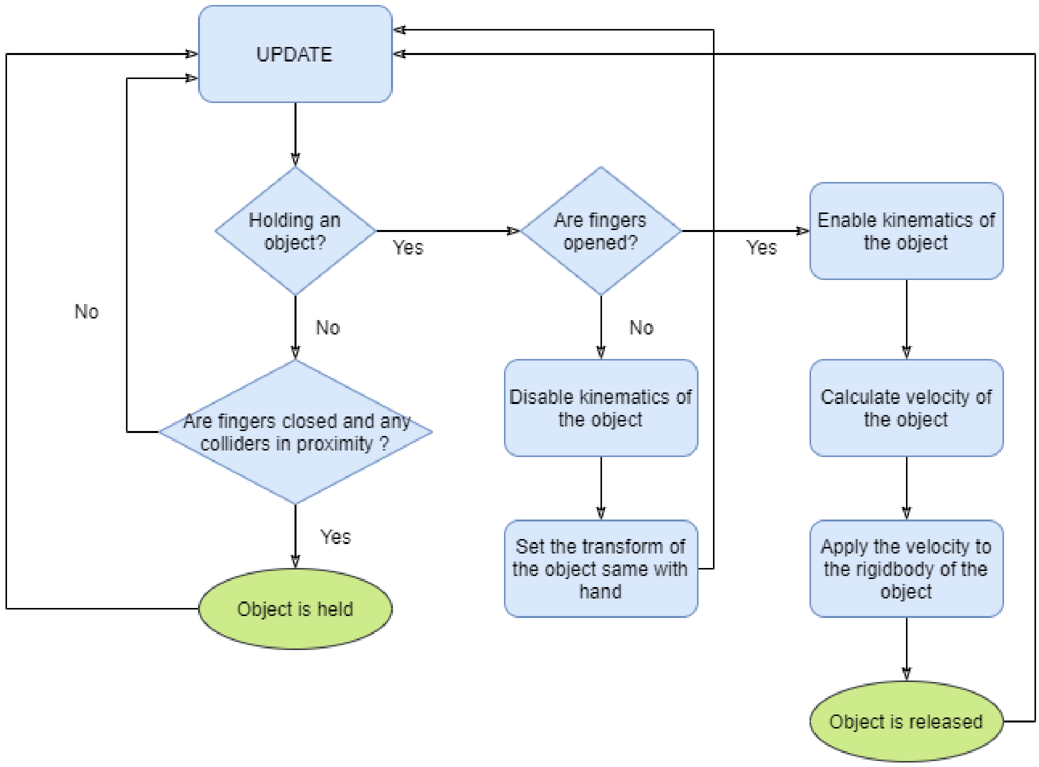


Figure 3.17: The Flowchart of the Grabbing and Releasing Interaction

```
{//set current object to the object we have picked up
  _currentObject = colliders[0].transform;
}
```

If the object is held, its kinematic is disable. This means the physic engine does not apply physics rules to that object any longer. Thus, the object moves with the hand model within the same transform. It can be considered as the object actually follows the transform, meaning the rotation and the position of the hand model as it is shown in Listing 3.7 (see in Fig. 3.18).

3.7: Manipulating the transform of the object with the hand

```
if (_currentObject != null)
{
  Rigidbody _objectRGB = _currentObject.GetComponent<Rigidbody>();
  _objectRGB.isKinematic = false;
  _currentObject.position = transform.position + ObjectGrabOffset;
  _currentObject.rotation = transform.rotation;
  _lastFramePosition = transform.position;
}
```

Figure 3.18: The hand is grabbing a cube

Case, the user's index finger curvature degree is smaller than 20 degrees while the object is held, the velocity of the object is calculated to be applied to it. The physics are enabled back so that the object can naturally fall down under gravity control. Listing 3.8 presents a part of the code of this functionality.

3.8: Releasing the object

```
if (angle <= 20 && _currentObject != null)
   {
      Rigidbody _objectRGB = _currentObject.GetComponent<Rigidbody>();
      _objectRGB.isKinematic = true;

   //calculate the hand's current velocity
      Vector3 CurrentVelocity = (transform.position - _lastFramePosition) /
          Time.deltaTime;

   //set the grabbed object's velocity to the current velocity of the hand
      _objectRGB.velocity = CurrentVelocity * ThrowMultiplier;

  //release the reference
      _currentObject = null;
  }
```

- **Manipulating a Door:**

In this interaction, the user is holding the knob and rotating it down until it unlocks. The knob should be rotated until 45 degrees to be able to manipulate the door. A 3D vector is calculated from the hand position point to the main joint of the knob. This vector is used in *Quaternion.LookRotation* function as forwarding vector to obtain a rotation along this forward vector. By using of *Quaternion.Lerp* function to interpolate transform, smooth rotation animation is achieved.

In the design of the door, a vertical beam is created in the joint of the door, which has the hinge joint attached to the door frame. The main door is created as a child object of this vertical beam, which means the main door rotation follows the beam's rotation. Likewise in the knob, a vector from the hand position to the beam position is calculated. Afterward, the angle between this vector and the forward vector of the beam is calculated to be applied to the angular velocity of the beam's rigidbody component to provide the door rotation along the hinge (see in Fig. 3.19).

Figure 3.19: The hand is opening the door

So far, the interaction methods described earlier are based on detecting colliders and gluing graspable objects, for example, a handle to the hand. Different methods are used in the following interaction examples to achieve a more natural and precise interaction.

- **Manipulating a Drawer:**

In this and the following interaction examples, the physics engine of Unity 3D plays an important role. As stated earlier, Unity 3D's *RigidBody* and *Collider* components cause collisions between 3D models. Using this feature, a sphere collider is placed at each fingertip. Similarly, each finger is hardened by adding the *RigidBody* component. Unlike the other examples described, fingers cannot enter the object in this interaction. They can push and pull some type of objects, just like in the real world. A good example of this interaction is opening a drawer. Likewise the fingers, the drawer handle contains

RigidBody and *Collider* components. So that, the fingers are able to push and pull the handle as it is shown in Fig. 3.20.

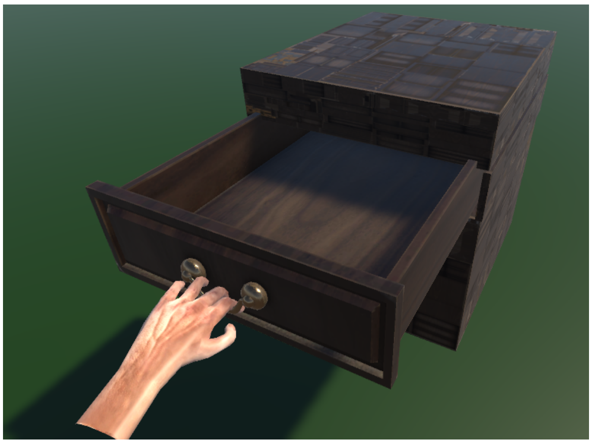


Figure 3.20: The hand is pulling the handle of a drawer

- **Playing a Piano:**

Going one step further, a piano playing demo is created involving a much more precise and delicate interaction. This interaction focuses on the fingers. As mentioned in the drawer example, each finger has *RigidBody* and *Collider* components on its tips. Therefore, all of the fingers are able to play notes. Likewise, each key of the piano contains these components. By setting the collision distance to a minimum, a precise touch level is obtained (see in Fig. 3.21). An approach similar to that rotating the knob of the door down is used to ensure that each key takes its starting position smoothly after the finger leaves the key. Each key contains *AudioSource* component with its own note sound clip. By using the OnTriggerEnter function, whenever a collision occurs between fingers and keys, each key plays its own notes once, as shown in the code presented in Listing 3.9.

3.9: Playing a note clip on each key of the piano

```
void onTriggerEnter()
{
   if(!play)
       {
          Audiosource.PlayOneShot(clip);
          play = true;
       }
}
```

Playing the piano and pushing/pulling an object, like in the Drawer example, has a fundamental advantage over the standard controllers used in VR systems. They provide precise interaction with fingers where a normal controller cannot perform these tasks, except those with finger detection. These interactions clearly demonstrate the usability and naturalness of the *CyberGlove-HT* system.

Figure 3.21: Playing piano

- **Interacting with a Menu:**

All interaction tasks shown so far are with 3D objects. An interactive menu has been designed to take advantage of the tip force sensors on the *CyberGlove-HT* and to implement the interaction between fingers and 2D user interface (UI) elements. A script to open the menu checks if the index and thumb fingers are touching by reading the force sensor values. If they are, a 2D menu will appear in front of the user and its position and orientation changes with relation to the scene camera in Unity 3D. Likewise, the user can close the menu at any time by touching his middle and thumb fingers to each other.

Each 2D button of the menu contains *Colliders.* The *OnTriggerEnter* function is used to detect collision starts and ends. When a collision occurs between the index finger and a button, the button calls the corresponding function and the button turns green until the finger is pressed the button again. These related functions can activate the right/left hand depending on which hand the user is wearing *CyberGlove-HT*, or any interaction task the user wants to perform. For example, if the user wants to play the piano, he/she can activate the colliders on the tips of the fingers. If he/she wants to grab a glass, he/she can deactivate these colliders in order to not to push the glass (see in Fig. 3.22).

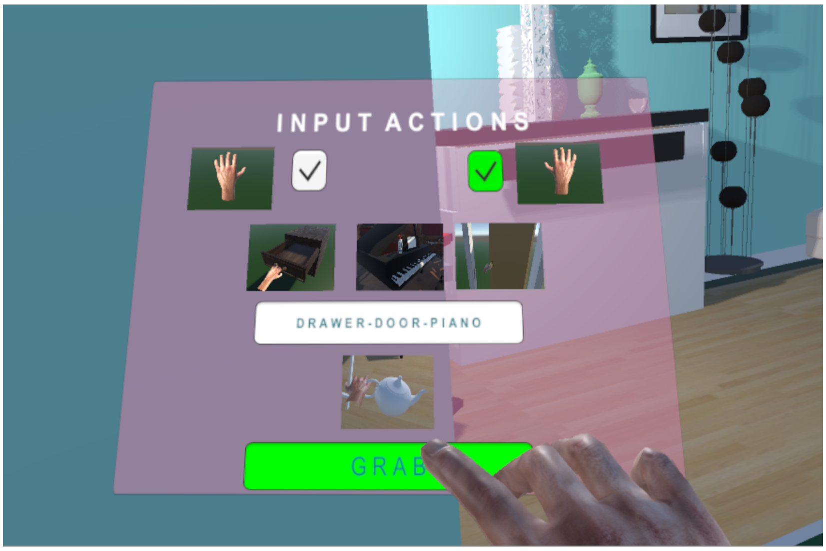


Figure 3.22: The user is activating the Grabbing mode via the Interactive Menu

- **Haptic and Visual Feedback:**

To increase the immersive level and the precise level, it is important to let the user know that they are colliding, touching an object in the virtual environment. Similar to HTC Vive Controllers, haptic feedback is provided from the virtual world to *CyberGlove-HT* to vibrate the glove each time the user grabs an object. Similarly, the object is highlighted to show the user that he can capture the object. *OnTriggerEnter* function detects the collision with hand and graspable object. Whenever it occurs, a boolean value *touch* is set as true so that the *CyberGlove-HT Receiver Script* access this information and sends *"1"* to *CyberGlove-HT* as described in section 3.2.2. At the same time, due to this collision the script accesses the RGB data of the object's material component to highlight the color. Afterwards, the ending of the collision is detected by *OnTriggerExit* function. The *touch* is assigned as false and the color of the object turns back to the initial color (see in Fig. 3.23).

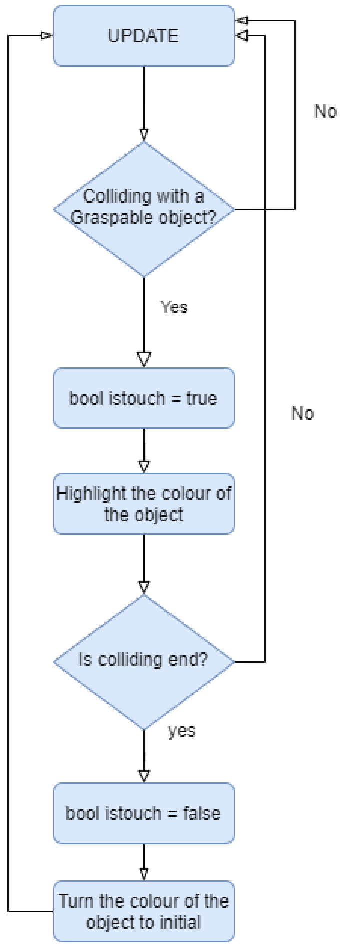


Figure 3.23: The flowchart of providing visual feedback

3.2.5 Integration of Kinect System

This section describes the implementation of the Kinect System. As mentioned earlier, this work is done in the context of the HTPDIR Project which proposes a low-cost system to map static and dynamic obstacles in the physical space letting users be aware of the limitations in the real world while he or she experiences the virtual world. It is a real-time scene 3D reconstruction module. The RGB-D sensors are located at a height of 2.20m and arranged to create a 4m x 4m quadrangular scenario as shown in Fig. 3.24. For the developed Unity scenario the reconstruction module provides the vertices' positions of a bounding box for each human and object in the workspace. Unity uses this information to render the bounding box meshes dynamically, in real-time. The next sections will detail the context of this information and how it can be used to represent people and objects in an immersive environment.

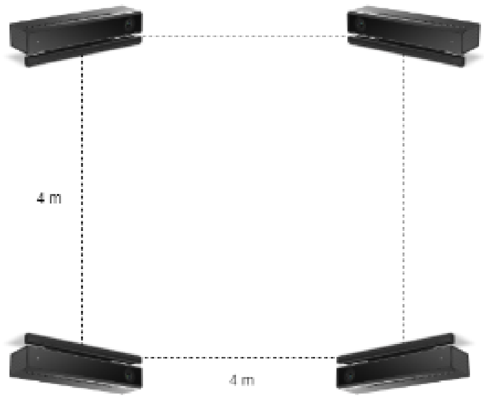


Figure 3.24: The Kinect quadrangular scenario

3.2.5.1 Communication and Data Parsing

The communication between Kinect Server and Unity 3D relies on UDP protocol as well as the communication with *CyberGlove-HT*. As explained in section 3.2.2, this process takes place in another parallel thread of the Unity 3D application.

Kinect Server provides position information of bounding box corners of tracked humans and objects [3] (See in Fig. 3.25). The calibration was carried out to calculate the *affine transformation matrix* to get the correct registration between the Kinect System and the Unity's Virtual World reference system. The reconstruction module provided 8 vertices positions for each human and objects are in Vırtual World. This information is forwarded to Unity 3D in *JSON* format. An example of the provided data is shown in Listing 3.10.

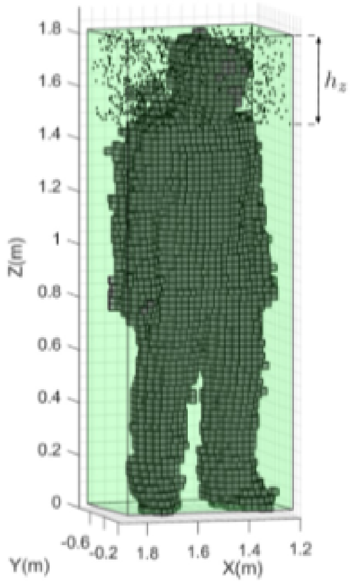


Figure 3.25: The Boundingbox of a human [3]

3.10: Data provided by Kinect Server


```
{
  "Players": [{
    "ID": 1,
    "HeadPosition": [0.00 0.00 0.00],
    "BoundingBox": {
      "p1": [0.00 0.00 0.00],
      "p2": [0.00 0.00 0.00],
      "p3": [0.00 0.00 0.00],
      "p4": [0.00 0.00 0.00],
      "p5": [0.00 0.00 0.00],
      "p6": [0.00 0.00 0.00],
      "p7": [0.00 0.00 0.00],
      "p8": [0.00 0.00 0.00]
    }
  }],
  "Objects": [{
    "ID": 2,
    "BoundingBox": {
      "p1": [0.00 0.00 0.00],
      "p2": [0.00 0.00 0.00],
      "p3": [0.00 0.00 0.00],
      "p4": [0.00 0.00 0.00],
      "p5": [0.00 0.00 0.00],
```

```
        "p6": [0.00 0.00 0.00],
        "p7": [0.00 0.00 0.00],
        "p8": [0.00 0.00 0.00]
      }

   }]
}
```


In order to parse this received *JSON* formatted data, first, the classes are created according to the structure of the data as in the following. A BoundingBox class is created with the vertices positions. Then, the Player class is created as it involves ID, Headposition and early created BoundingBox elements. Likewise, Object and Item classes are created with specified elements in the data structure. Finally, RootObject class, which contains the Item class is created to be used in parsing as in the following code presented in Listing 3.11.

3.11: Creating JSON classes

```
public class BoundingBox
{
    public List<float> p1;
    public List<float> p2;
    public List<float> p3;
    public List<float> p4;
    public List<float> p5;
    public List<float> p6;
    public List<float> p7;
    public List<float> p8;
}
public class Player
{
    public int ID;
    public List<float> HeadPosition;
    public List<BoundingBox> BoundingBox;
}
public class Object
{
    public int ID;
    public List<BoundingBox> BoundingBox;
}
```

```
public class Item
{
    public List<Player> Players;
    public List<Object> Objects;
}
public class RootObject
{
    public List<Item> Items;
}
```

Afterward, the received data is parsed by using Unity 3D's *JsonUtility* class as shown in Listing 3.12.

3.12: Parsing JSON data

```
RootObject parsed_data = JsonUtility.FromJson<RootObject>(jsonData);
```

.

3.2.5.2 Human and Object Tracking Visualization

The parsed data provides the bounding boxes vertices coordinates for each human and object in the workspace. By using these 8 points, cubes and quadrangular prisms meshes are created individually with random colors for each object and human, except the human with the HTC-Vive headset. Therefore, the user with the headset is able to see around the immersive environment. After creating the bounding boxes (i.e. cubes and quadrangular prisms), their position is updated according to the newly received data. In the end, human and object tracking visualization in the immersive environment process is archived, creating a mixed reality (see in Fig. 3.26 and Fig. 3.27).

Meshes are created from points defined in 3D space which are called as vertices. We are able to create meshes dynamically by connecting three vertices to obtain a triangle Unity 3D scripts by using *MeshFilter* and *MeshRenderer* components. The steps of Mesh generation using triangles is described as follows:

- Define the cube's dimensions,
- Define the each Corner coordinates ,

- Define the vertices as Bottom, Left, Front, Back, Right and Top,
- Define each vertex's Normal,
- Define the triangles that make up the Mesh,
- Build the mesh.

i

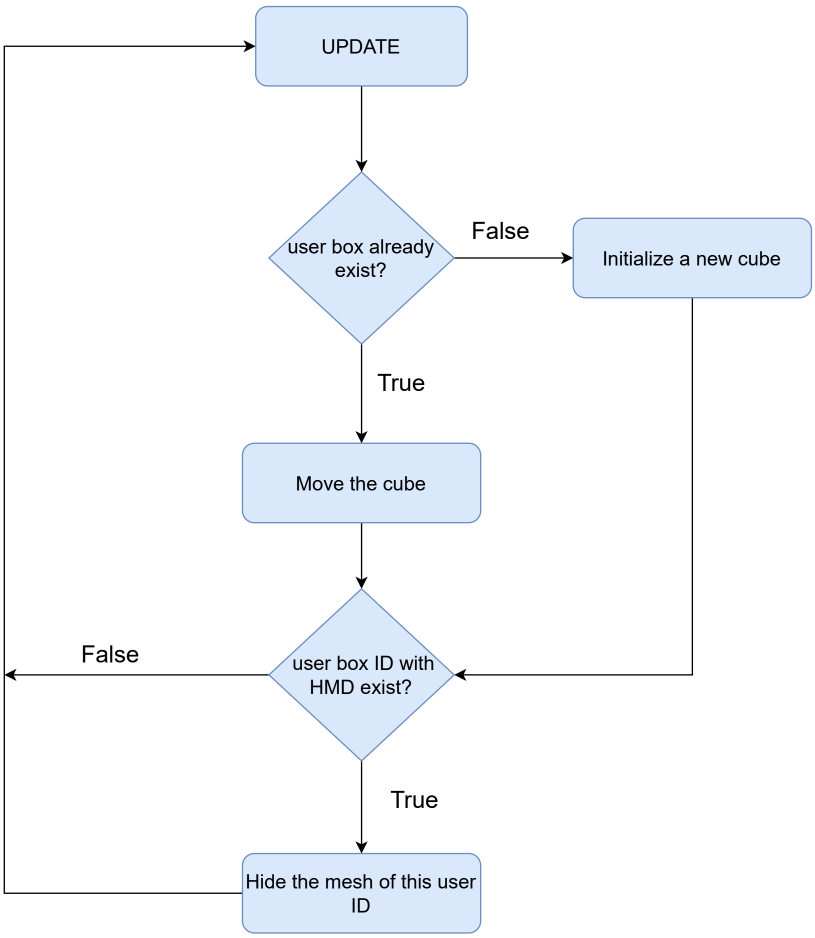


Figure 3.26: The flowchart of Human and Object Tracking Visualization

To move the cubes to their new location, a new mesh is created with the described algorithm. The central location of this mesh is checked to see if it is the same as before. If not, the cube transformation is set to the new mesh center location as in Listing 3.13.

3.13: The translation of the cubes

```
Vector3 new_position = newmesh.bounds.center;
```

```
if (playerCube.transform.position != new_position)
   {
     playerCube.transform.position += (new_position -
         playerCube.transform.position) * velocity * Time.deltaTime;
    }
```

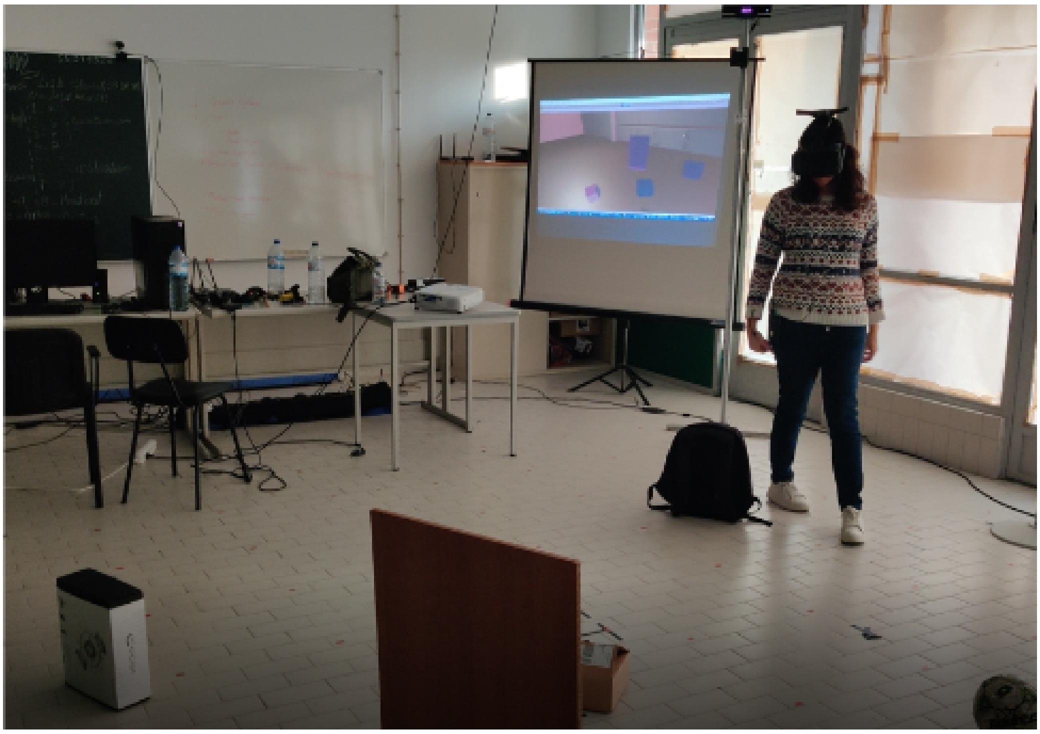


Figure 3.27: Virtual world built from the real world (image shown in projection screen). The Kinect sensor-based reconstruction system maps the physical elements as bounding boxes.

Chapter 4

Gesture Recognition Based on Inertial Sensors

In this chapter, the proposed architecture to classify hand gestures is described. In Section 4.1 we describe the Matlab framework used for processing data and also for building the classification architecture in order to recognize horizontal and vertical swipes and static(with no movement) gestures. This chapter also describes the overall methodology which includes, data collection (Subsection 4.1.1), classification approach (Subsection 4.1.2) and lastly the Communication with Unity (Subsection 4.1.3).

4.1 Matlab framework for IMU based gesture recognition

Matlab[1] was used as a prototyping tool to:

- Analyze IMU quaternion (x,y,z,w) data of the CyberGlove-HT,
- Develop classification algorithms for gesture recognition,
- Train a network model with these data and use the trained model in online/offline applications.

Our approach uses the *CyberGlove-HT* wearable sensory glove system for collecting data for tracking users hand motion.

Figure 4.1, shows the steps of the gesture recognition approach which are described below.

[1] https://www.mathworks.com/

Figure 4.1: The Adopted Methodology of Imu-Based Gesture Recognition

4.1.1 Data Collection

The data acquisition was made during short recording periods for three different gestures namely:

- **Static:** Where the user does not move his/her hand;
- **Horizontal Swipe:** Where the user swipes move his/her hand in horizontal axis;
- **Vertical Swipe:** Where the user swipes move his/her hand in the vertical axis.

While wearing the *CyberGlove-HT* the subject performed multiple times each gesture in every direction (360 degrees). Each record was taken approximately in 1-3 seconds. There is downsampling in the communication between the CyberGlove-HT and Matlab. The reason is for Matlab, to receive data and classify it at a larger rate is harder. In the end of experimentally trying 20 Hz was ideal for Matlab to deal with real-time data collection and to provide reliable classification results. Therefore, the data were collected every 50 ms using the same communication infrastructure as described in chapter 3.2.2, i.e., IEEE 802.11 Wi-Fi and UDP communication protocol. The *CyberGlove-HT* sends the packets in string type. MATLAB receives this string data and parses it by using ','. Parsed data are treated as floats (format sent from Cyberglove-HT). Within Section 3.2.2, it was described the packet format that received from *CyberGlove-HT* and its specifications in table 3.1. This packet includes quaternion values of IMU which is the output of sensor fusion of the rotation information obtained from the inertial sensors such as accelerometer, gyroscope and magnetometer (see in Fig. 4.2). The gesture recognition, use this quaternion information is used as a feature vector which is shown in Eq. 4.1 where q is the quaternion. Quaternions provide a convenient mathematical notation for representing space orientations and rotations of objects in three dimensions which is represented with complex numbers, real number (q_w) and imaginary numbers (q_x, q_y and q_z).

$$Q = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix} \tag{4.1}$$

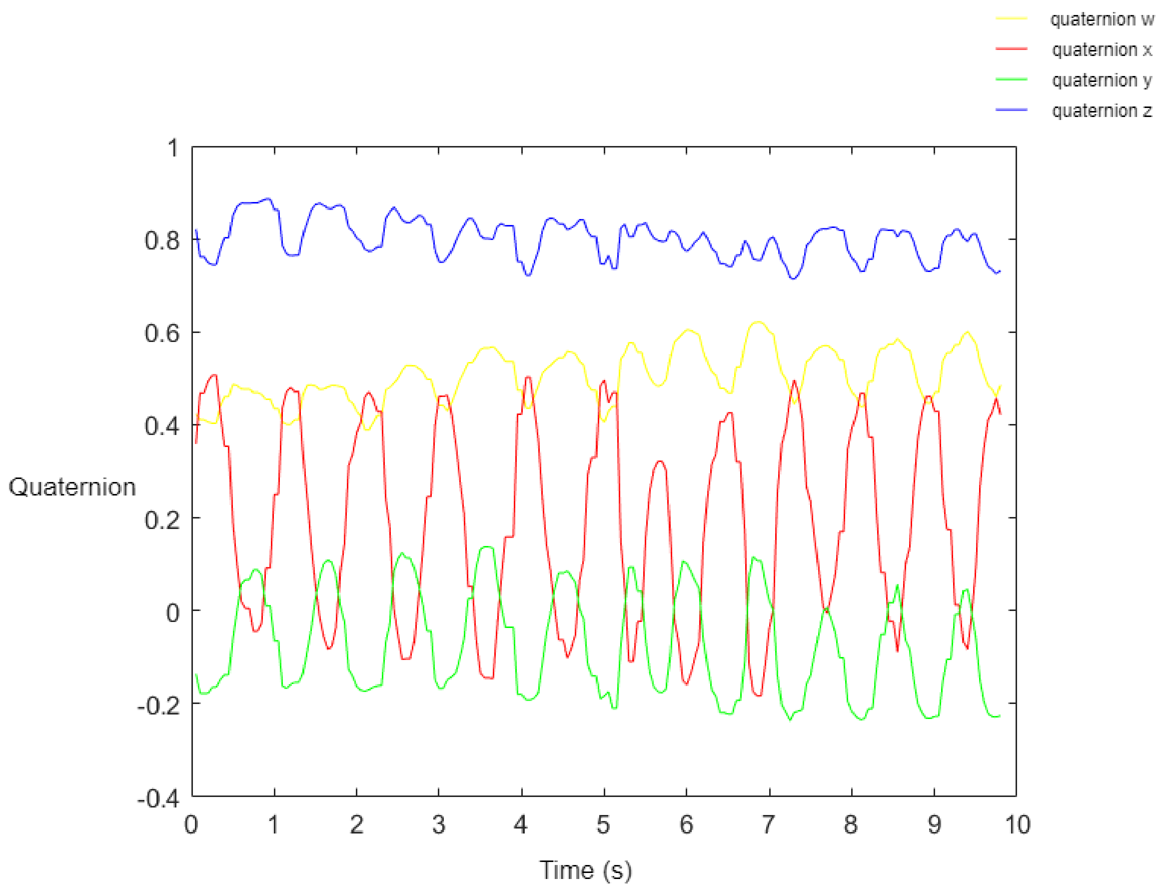


Figure 4.2: The Quaternion Signals received from CyberGlove-HT while the user performs vertical swipe gesture

After the data is being parsed, each quaternion value of a given time sample is was introduced in a cell array and its correspondent category in a categorical array. These categorical arrays indicate the labels that correspond to the gestures respectively. For *Static*, the label is *'0'*, for *Horizontal Swipe*, it is *'1'* and for *Vertical Swipe*, it is *'2'*.

The recorded dataset includes 1100 trials of gestures. The distribution of these trials is shown in table 4.1.

Table 4.1: Number of trials for each class

Activities	***Number of trials***
Static	594
Horizontal Swipe	254
Vertical Swipe	252

When preparing the data, the number of samples was taken into consideration as well as the time of each sample ($\sim$ 2 seconds each). The average number and standard deviation of samples for each class data are indicated in table 4.2.

Table 4.2: Number of samples for each class

Activities	***Avg of samples - time (sec.)***	***Std of samples***
Static	26.44 - 1.32	4.75
Horizontal Swipe	25.03 - 1.25	5.41
Vertical Swipe	25.06 - 1.25	5.43

4.1.2 Classification Approach

After preparing the data set, we proceeded to training and testing of the proposed framework to classify motions as static hand, horizontal swipe and vertical swipe. 80% / 20% of the data set selected randomly to be used in training and testing respectively, as shown in 4.3. After some preliminary testing, we used the *Deep Learning Toolbox* from Matlab in order to employ the LSTM methodology. The LSTM hyperparameters were set as follows: minibatch size as 88 (approximately 10% of the total number of training data), the maximum number of the epoch is 500 and the learning rate initially set at 0.01. The window classification size is 40 samples which corresponds to a time window of 2 seconds. In real-time classification, the last 40 samples of the received data packet are being fed to the classifier. In offline classification, each trial is classified with variable sample size, manually acquired and labeled. Put it differently, offline acquired trials (gestures) may have different time samples, depending on the user input. The input vector, be it for real-time or offline classification, corresponds to the quaternion information shown in formalization 4.1. This means that the LSTM is fed with a set of four features, regardless of the number of samples. This is possible since LSTM layers are able to consume variable-length inputs and ultimately produce only the layer's output at the final sequential step. The decision rate is 20 Hz, every 50 ms, the network provides an output. The method is synchronous when training and testing the network, using a number of samples that would depend on the action being classified, with a starting point and end point defined by the user. During real time analysis, the method is asynchronous, running at 20Hz over a sliding window. This also implies that there is an overlapping in the classification window as it is shown in Fig. 4.3 .

All this general classification approach is better illustrated in Fig. 4.4. A four dimensional feature data, $x_t = \begin{bmatrix} q_w(t) & q_x(t) & q_y(t) & q_z(t) \end{bmatrix}$ with a variable sequence

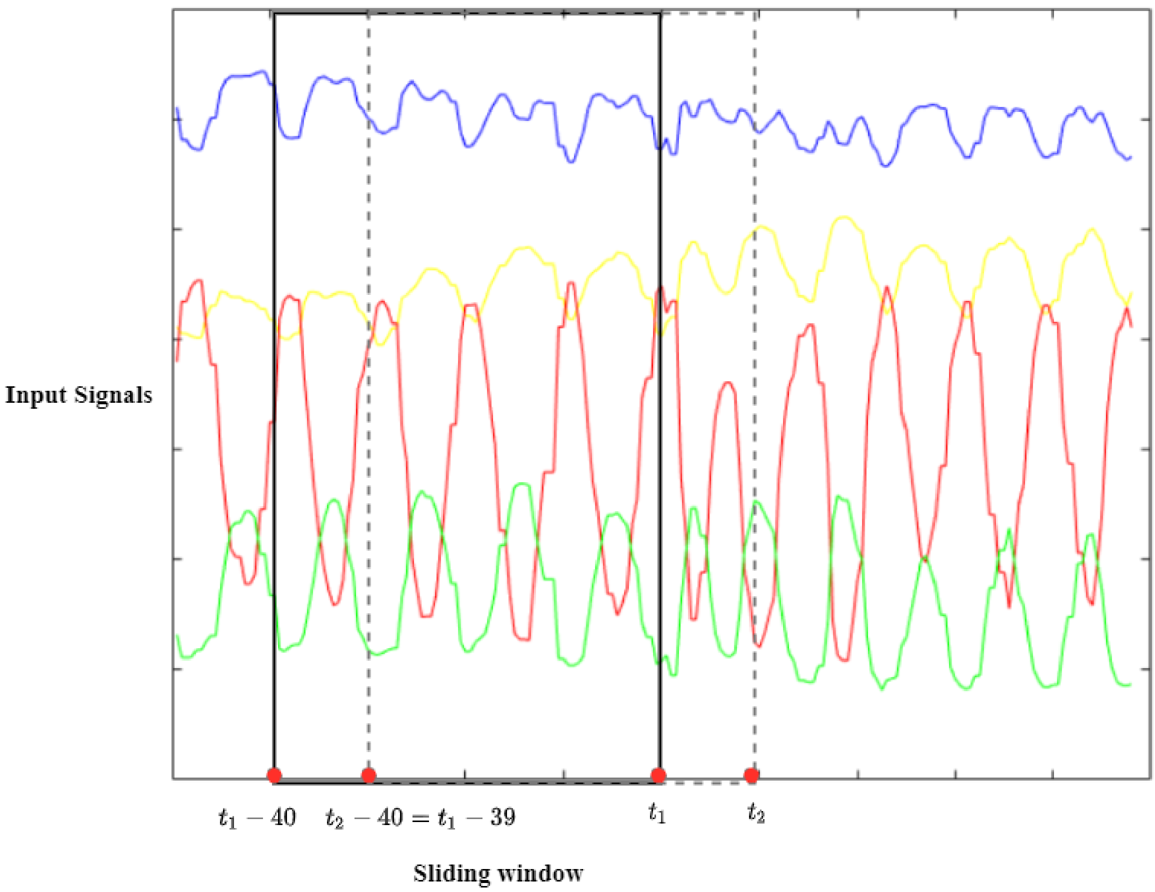


Figure 4.3: The sliding classification window

length,is the input vector of LTSTM, n is the number of total trial and M_n is the number of samples of each trial that feeds offline training and testing. In case of run-time classification M_n is 40.

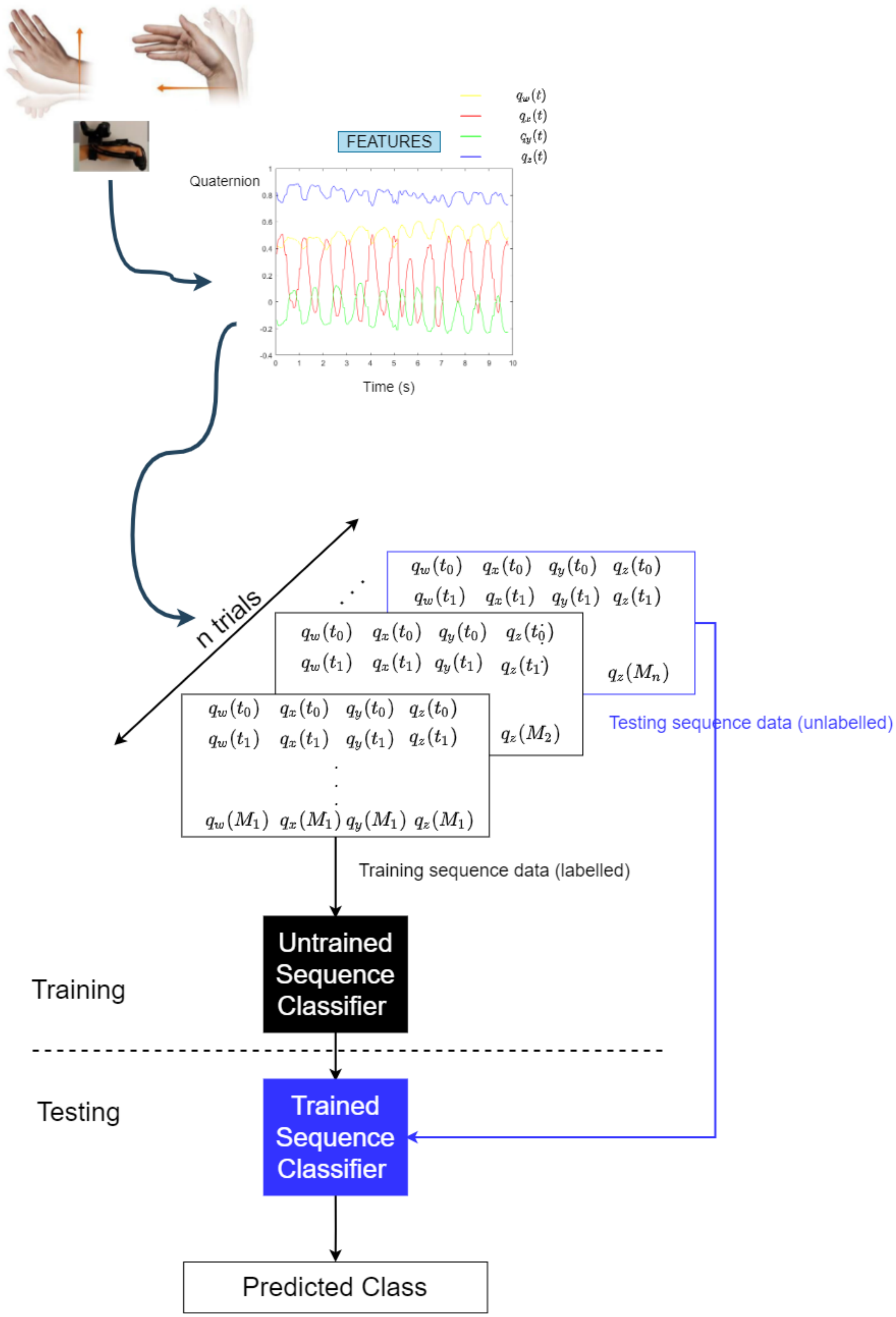


Figure 4.4: General classification approach applied to the hand motion (The hand figures are taken from [4]).

Table 4.3: Number of trials for training and test

Porpuses	***Number of data***
Training	880
Testing	220

4.1.3 Implementation of Gesture Recognition in Unity

This section explains how to implement the hand gesture recognition feature in Unity to achieve a new interaction method. A new scrollable menu has been designed for this purpose. In this method, the user can scroll the menu up and left by swiping his/her hand in the respective directions.

The followed approach includes three main components (see in Fig. 4.5) :

- CyberGlove-HT, in order to collect the quaternion data of the IMU of the glove for each gesture;
- Matlab, to perform gesture recognition and send the gesture information to Unity 3D;
- Unity 3D, to receive the gesture information data and swipe a menu (see in Fig. 4.6).

In Matlab, the result of the LSTM provides the class of the performed gesture. This data is encoded in JSON string format. Afterward, the JSON data is sent by using the UDP protocol with a frequency of 50 Hz. In Unity, firstly, the data is received in JSON format in the same way described in Section . Afterward, it is decoded and stored as a string value. This string value is used in a Logic Block that checks the data corresponding to which gesture. If the data corresponds to the horizontal swipe, the menu will be scrolled in horizontal axes once. In the same way, if it corresponds to vertical axes, the menu will be scrolled in the up direction.

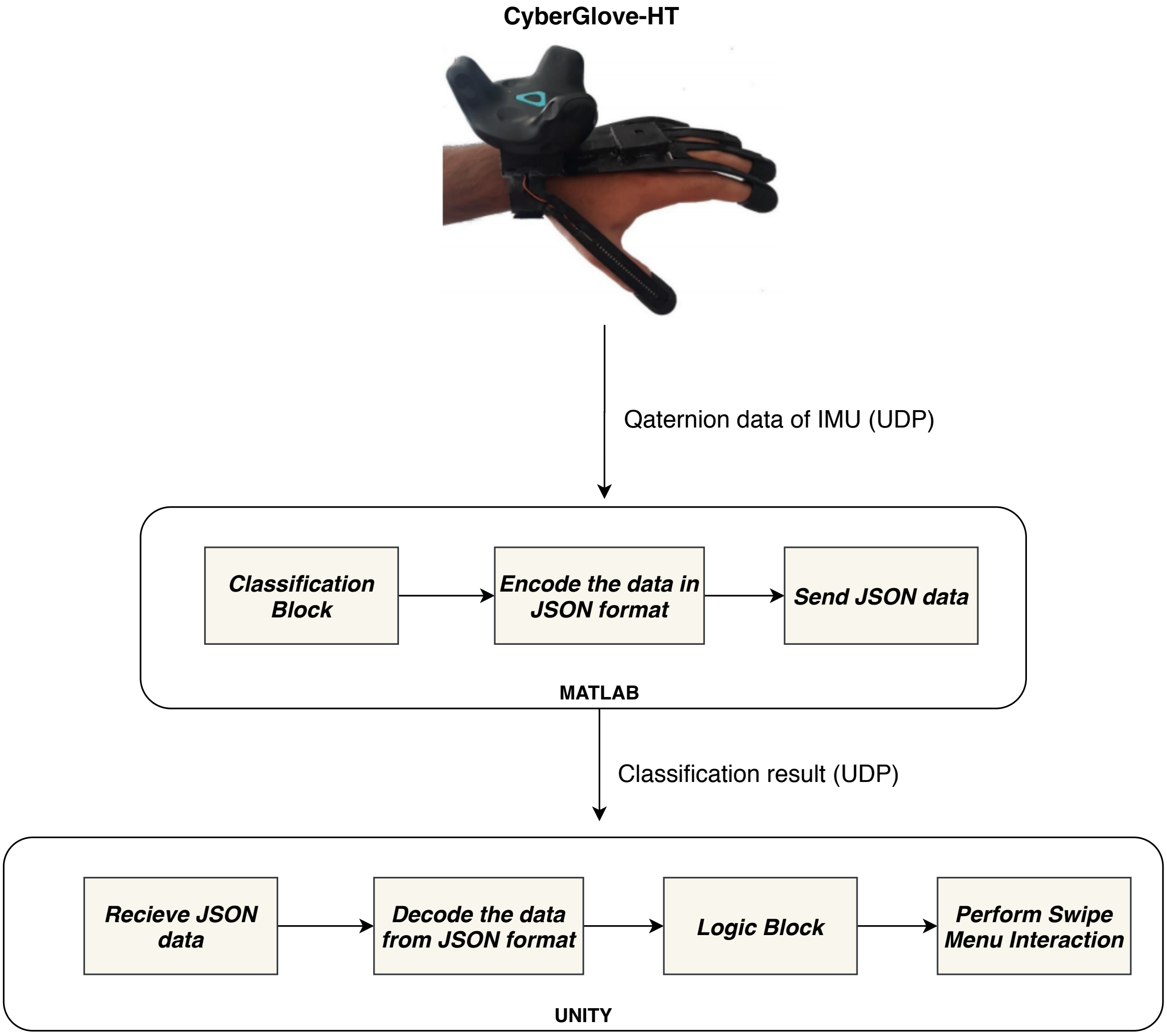


Figure 4.5: The Approach of Communication between Unity and MATLAB

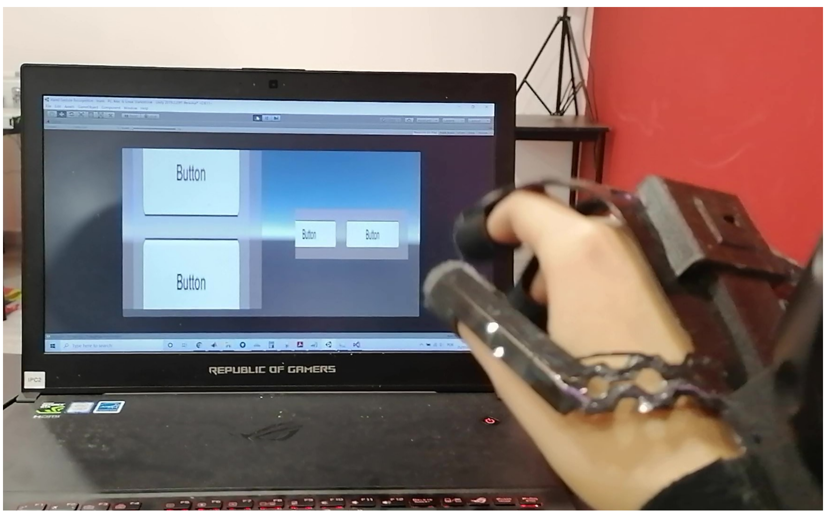


Figure 4.6: The user swipes the menu up with swiping the hand vertically

Chapter 5

Experimental Results

This chapter describes the proposed evaluation methodology for the developed VR application and devices in terms of naturalness, usability, immersion, embodiment and task performance. Furthermore, the quantitative evaluation methodology for the performance of IMU-based gesture recognition is provided.

5.1 Interaction Tasks in Immersive Environment

This section describes the proposed experimental design and procedure, and the naturalness, immersion, embodiment and task performance results for developed VR application. The experiments aim to assess if glove-based systems can contribute to a higher sense of immersion, embodiment and usability when compared to standard VR hand controller devices (typically button-based).

5.1.1 Experimental Design and Procedure

In this experimental study, a typical task commonly performed at home has been designed where a person crosses the several divisions of a house and for this, he/she has to open a door and transpose it. Since embodiment and realism of the movement of the hand during the handle rotation are the main focus, the detection of movement of the wrist and hand are essential. A virtual door was designed as described in Section 3.2.4. Each person, wearing a VR HMD system, disposing of an egocentric view, was invited to perform the following door opening based sub-tasks as it is shown in Fig. 5.1:

- A - Walk to the door;
- B - Unlock door (rotation of handle);

- C - Push the door;
- D - Pass through the door.

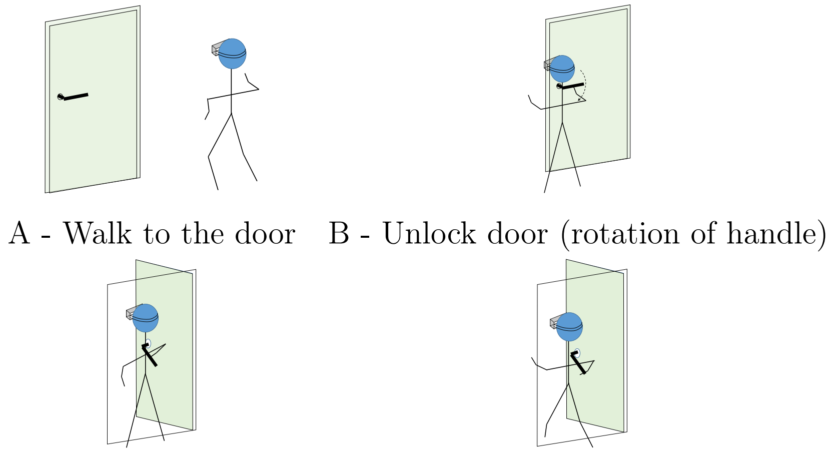


Figure 5.1: Sub-tasks to open the door in VR environment.

These tasks were repeated by each subject 2 times. In the first trial, users are holding the standard HTC Vive hand controller that has a click button interaction and provides position and orientation. In order to grab the door's handle user had to press a button. In the second trial, users are wearing the developed CyberGlove-HT, which frees the hand and enables grab functionalities. To grab/release the door's handle, the user had to close/open the fingers.

Twenty-two participants (7 women and 15 men) from the Polytechnic Institute of Tomar were invited to test the system. Participants were mainly students and researchers from engineering courses. Participants were aged between 20 to 46 years old (μ=26.56, α=6.66). and its participation was voluntary. Four of the 22 participants never had contact with video games technology and only 3 subjects had previous experience with interaction devices like the proposed CyberGlove-HT. Subjects were invited individually to the lab and informed about the procedure to open the virtual door while a software application recorded the performance measures during task execution. The experiments involving the HTC Vive controller and the CyberGlove-HT were performed randomly and at the end, users filled subjective questionnaires.

Qualitative and quantitative measures were obtained for Efficiency measures and Immersion and presence measurements. Efficiency measures are:

- Time - measures the time taken by a person to open a door and pass through it:

  - Total time: overall time to accomplish the task,
  - Sub-task time: measure the time taken by a person in each door opening tasks (A, B, C and D),

- Length: length of the path described by the hand in each sub-tasks.

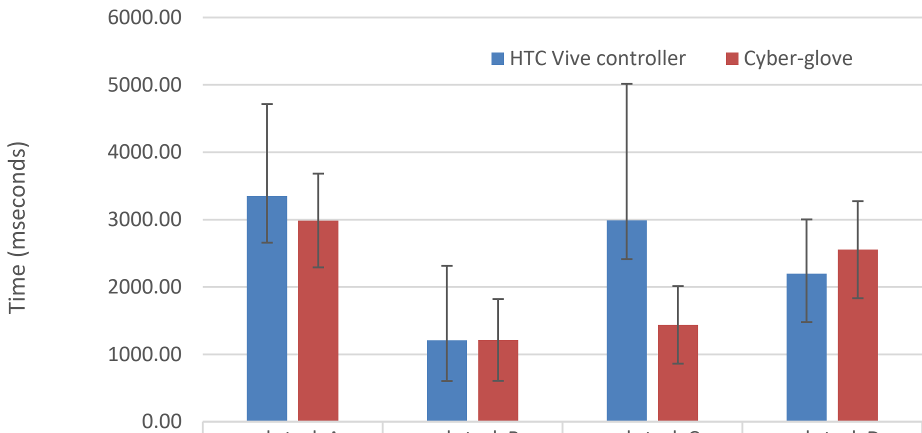
In order to evaluate the immersion and presence experience qualitatively, the subjects were invited to fill a questionnaire on a scale from 1 to 7, where 1 meant very weakly and 7 very strongly, enabling to determine issues like naturalness (question Q1), usability (question Q2), immersion feeling (question Q3) and sense of embodiment (question Q4). These questions were adapted from IBM Computer Usability Satisfaction Questionnaire [44] and from Usoh and Slater Presence Questionnaire [45].

- Q1 - How natural was the interaction with the VR environment?
- Q2 - How easy was manipulating and moving objects in the VR environment?
- Q3 - How strongly was the immersion feeling in the VR environment?
- Q4 - Did I feel that my own hand was manipulating and moving objects in the VR environment?

5.1.2 Results and Discussion

Figure 5.2 indicates the mean of task-time performances in each sub-tasks of the virtual door opening task, using the HTC Vive controller or the CyberGlove-HT. Results show that that in sub-task C users pushed the door faster wearing the CyberGlove-HT. Users reported that releasing the door's handler was easier with the cyber-glove because they just had to open the hand/fingers, and they were not concerned about the instant to release the HTC controller button. Thus, users performed sub-task C better with the cyber-glove, being both task-time and hand's length path statistically significant.

Figure 5.3 presents the total mean task-time results, and the total mean length of the path described by the hand of participants. The time and length of the global task are smaller for the CyberGlove-HT. Metric results revealed that 83% of the users performed faster door pushes, and described shorter paths with their hands wearing the CyberGlove-HT.



	sub-task A	sub-task B	sub-task C	sub-task D
HTC Vive controller	3352.98	1210.83	2989.09	2199.14
Cyber-glove	2986.73	1213.22	1437.38	2553.24

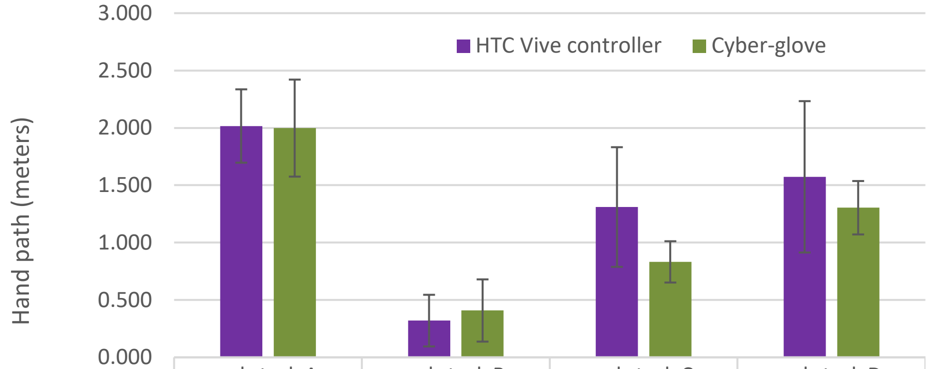
Figure 5.2: Mean task-time performance of participants for each sub-task

Figure 5.4 shows the mean length of the path described by the hand in each sub-tasks of the virtual door opening task, using the HTC Vive controller or the CyberGlove-HT.

Figure 5.5 indicates the results of the questionnaire. According to the results,

a) b)

Figure 5.3: a) Total mean task-time, and b) total mean length of the path described by the hand of participants while performing the virtual door opening global task, HTC Vive controller vs cyber-glove.



	sub-task A	sub-task B	sub-task C	sub-task D
HTC Vive controller	2.016	0.320	1.310	1.574
Cyber-glove	1.998	0.408	0.832	1.304

Figure 5.4: Mean length of the path described by the hand of participants, for each sub-task

100% of the participants rated the CyberGlove-HT based interactions as equally or more natural, and 90% of users experienced an equal or a significant increase in the

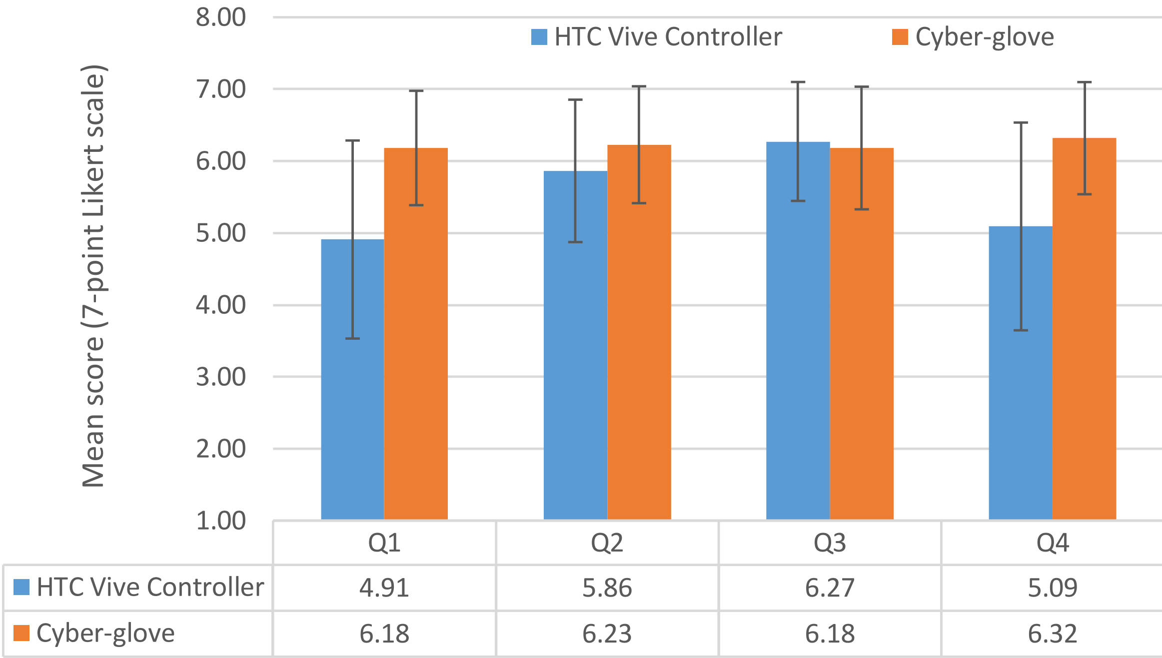


Figure 5.5: Results of subjective questionnaires

sense of embodiment.

Qualitative evaluation based on questionnaires to the users shows that the CyberGlove-HT highly contributes to naturalness. Factors like usability and immersion feeling seem similar for both devices, and the sense of embodiment is significantly improved with the CyberGlove-HT. Several participants reported that with the CyberGlove-HT they had more freedom of movement, while with the HTC Vive controller they felt hand movement constraints, and were afraid of dropping the controller during the release of the door handle.

5.2 IMU-Based Gesture Recognition

As described in Chapter 4, gesture recognition allows three different gestures (Static, Horizontal Swipe and Vertical Swipe) to be recognized. To visualize the algorithm performance and since it was used a supervised method, we have the confusion matrix as it is shown in table 5.1 That matrix gives us the values of TP, TN, FN and FP where, TP = True Positive, TN = True Negative, FN = False Negative and FP= False Positive. True positives correspond to samples that were correctly classified, False positives are the ones that were incorrectly classified. Similarly, True negatives are the samples that were correctly classified as negatives and False negatives are the ones that were wrongly classified as negatives.

Table 5.1: Confusion Matrix

		Output	
		Positive	Negative
Target	Positive	TP	FP
	Negative	FN	TN

The offline result is shown in Fig. 5.6 as a Confusion Matrix which are obtained with collected datasets, still the algorithms were also tested in real-time. The accuracy of the model performs 100%, slightly high in *Horizontal Swipe* gesture (*Class 1*) and *Vertical Swipe* gesture (*Class 2*). Even though *Static* gesture (*Class 0*) does not perform as very highly as *Class 1* and *Class 2*, still the model performance is promising.

Figure 5.6: The confusion matrix of the offline results

Chapter 6

Conclusion

One of the main challenges in Immersive technologies, such as Virtual Reality (VR), is to provide to the user Immersion and its physicological result Presence, i.e feeling of "being there". This is the essential condition of any Immersive Environment application which relies on convincing the user to be in an alternative world. A world where the user is able to behave in the same way in daily life. Many research domains such as psychology, employes Immersive technologies in order to investigate human behaviors in various different Immersive scenarios by relying on this principle. Thus, improving the immersive level in Virtual Reality applications has been the main field for many researchers and developers. With this motivation in mind, the main aim of this project focused on the creation of an Immersive Environment system capable of providing physical interactions performed by a user wearing a cyber glove (CyberGlove-HT). The user is allowed to manipulate objects such as grab/ release an object, open and close doors, and drawers, playing piano, with his/her own hand instead of using any VR controller. As physical interactions enhance the Immersive level, likewise, naturalness in self-embodiment plays a huge role in contributing to the Immersive level by providing a match between visualization and the real body. Thus, in this project, the user is able to manipulate his/her own hand the same way in the real world such as rotation of the hand and wrist, open and close the fingers. Furthermore, gesture recognition based on the inertial sensor of the CyberGlove-HT is presented which allows the user to manipulate objects such as swipeable menus, by recognizing three gestures, namely, swipe horizontal and vertical and no movement.

Chapter 5 presented the experimental study we made in order to compare the CyberGlove-HT with HTC controllers regarding contribution to naturalness, self embodiment, and presence. The participants were asked to open a door and transpose it as the same in daily life with CyberGlove-HT and with the HTC Controller. The results show that users pushed the door faster wearing the CyberGlove-HT. Users performed this sub-task better with CyberGlove-HT and they reported that releasing the door's handler was easier with the cyber-glove since they just had to open the hand/fingers, instead of releasing the HTC controller button. The time and length of the global task are smaller for the CyberGlove-HT. Metric results revealed that 83% of the users performed the faster door pushes and described shorter paths with their hands wearing the CyberGlove-HT. According to the results of questionnaires, 100% of the participants rated the CyberGlove-HT based interactions as equally or more natural and 90% of users experienced an equal or a significant increase in the sense of embodiment. The users rated the CyberGlove-HT hıghly contributes to naturalness and the sense of embodiment is significantly improved with the CyberGlove-HT.

In order to test quantitively the gesture recognition based on inertial sensors, data collected from CyberGlove-HT. LSTM method was used to classify these gestures in training. After preparing the data set, we proceed to train and test the proposed framework to classify motions as static hand, horizontal swipe, and vertical swipe. The quantitative offline results show that the accuracy of the model performs 100%, slightly high in horizontal and vertical swipes. Even though Class 0 does not perform 100%, it still performs quite high.

Future work includes improving the self embodiment of the hand model by allowing 6 DoF to fingers. The sink of the fingers in grabbing task still remains as a challange. Thus, further exploration is needed to enhance the naturalness level in grabbing tasks. More complex tasks in virtual and mixed reality will be carried out to validate the overall system.

Bibliography

[1] L. Almeida, E. Lopes, B. Yalçinkaya, R. Martins, A. Lopes, P. Menezes, and G. Pires, "Towards natural interaction in immersive reality with a cyber-glove," pp. 2653–2658, 2019.

[2] H. C. J. Paul Milgram, "A taxonomy of real and virtual world display integration," 1999.

[3] D. Bicho, P. Girão, J. Paulo, L. Garrote, U. J. Nunes, and P. Peixoto, "Markerless multi-view-based multi-user head tracking system for virtual reality applications," *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 2645–2652, 2019.

[4] S. Mohatta, R. Perla, G. Gupta, E. Hassan, and R. Hebbalaguppe, "Robust hand gestural interaction for smartphone based ar/vr applications," pp. 330–335, 2017.

[5] J. Jerald, "The vr book: Human-centered design for virtual reality," 2016.

[6] M. McCreery, P. Schrader, S. Krach, and R. Boone, "A sense of self: The role of presence in virtual environments," *Computers in Human Behavior*, vol. 29, p. 1635–1640, 07 2013.

[7] A. B. . B. F. Bohil, C., "Virtual reality in neuroscience research and therapy," *Nat Rev Neurosci 12, 752–762*, 2011.

[8] A. B. . B. F. Bohil, C., "Virtual reality in neuroscience research and therapy," *Nat Rev Neurosci 12, 752–762*, 2011.

[9] J. Steuer, "Defining virtual reality: Dimensions determining telepresence," *https://doi.org/10.1111/j.1460-2466.1992.tb00812.x*, 1992.

[10] C. Heeter, "Being there: The subjective experience of presence.," *Presence, 1:262–271.)*, 1992.

[11] T. Sheridan, "Musings on telepresence and virtual presence.," *Presence, 1:120–126.)*, 1992.

[12] D. Schloerb, "Being there: A quantitative measure of telepresence.," *Presence, 4:64–80.)*, 1995.

[13] G. Coates, "Program from invisible site—a virtual sho, a multimedia performance work presented by george coates performance works," 1992.

[14] P. Greenbaum, "The lawnmower man," *Film and video, 9 (3), pp. 58-62.*, 1992.

[15] J. T. Bell and H. S. Fogler, "The investigation and application of vr as an educational tool," *Proceedings of the American Society for Engineering Education*, 1995.

[16] S. Mandal, "Brief introduction of vr its challenges.," 2013.

[17] T. Mazuryk and M. Gervautz, "Vr-history, applications, technology and future.," 996.

[18] O. Bimber and R. Raskar, "Spatial augmented reality: merging real and virtual worlds.," *CRC Press*, 2005.

[19] J. H. S. S. Fisher, M. McGreevy and W. Robinett, "Virtual environment display system.," *i Proceedings of the 1986 workshop on Interactive 3D graphics, pp. 77–87*, 1987.

[20] Z. Zivkovic, "Optical-flow-driven gadgets for gaming user interface.," *Entertainment Computing-ICEC 2004, Springer, pp. 90–100*, 2004.

[21] M. Vafadar, "Vr: Opportunities and challenges.," *History of VR and its use in medicine*, 2013.

[22] M. K. J. Novak-Marcincin and K. A. Beloushy, "Vr technologies and virtual manufacturing in manufacturing engineering.," *Scientific Bulletin Series C: Fascicle Mechanics, Tribology, Machine Manufacturing Technology, vol. 23, no. 100*, 2009.

[23] P. N. F. Rebelo, E. Duarte and M. M. Soares, "Human factors and ergonomics in consumer product design: Methods and techniques," *24 VR in Consumer*, p. p. 381, 2011.

[24] M. Farshid, J. Paschen, T. Eriksson, and J. Kietzmann, "Go boldly! explore augmented reality (ar), virtual reality (vr), and mixed reality (mr) for business," *Business Horizons*, 07 2018.

[25] C. F. P. Q. Z. J. Song, H. and P. Gu, "Improvement of user experience using virtual reality in open-architecture product design.," *Proc. Inst. Mech. Eng. B J. Eng. Manufact*, p. 232, 2017.

[26] B. D. G. N. Schmidt, M. and C. Schmidt, ""a prototype immersive, multi-user 3d virtual learning environment for individuals with autism to learn social and life skills: a virtuoso dbr update,"," *International Conference on Immersive Learning, Cham: Springer*, p. 185–188, 2017.

[27] O. A. D. Englund, C. and L. Price, "Teaching with technology in higher education: understanding conceptual change and development in practice.," *High. Educ. Res. Dev.*, pp. 36, 73–87., 2017.

[28] R. E. M. C. H. H. G. F. M. P. M. G. Gallagher, A. G., "Virtual reality simulation for the operating room: proficiency-based training as a paradigm shift in surgical skills training.," *Ann. Surg.*, p. 241:364.

[29] F.- J. G. V. G.-P. A. Botella, C. and R. Baños, "Recent progress in virtual reality exposure therapy for phobias: a systematic review.," *Curr. Psychiatry Rep.*, p. 19:42, 2017.

[30] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst, "Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques," *Computer Graphics Forum*, 1998.

[31] J. P. S. Jonathan Lin, "Towards naturally grabbing and moving objects in vr," *IST International Symposium on Electronic Imaging, The Engineering Reality of Virtual Reality*, 2016.

[32] M. Moehring and B. Froehlich, "Effective manipulation of virtual objects within arm's reach.," *2011 IEEE Virtual Reality Conference, pp. 131–138, March*, 2011.

[33] Élio Leno da Costa Lopes, "Cyberglove para interação com ambientes imersivos," 2019.

[34] D. L. J. Z.-Y. M. P. Z. Yang Wenzhen, Wu Xinli, "A master-slave hand system for virtual reality interaction," *International Conference on Virtual Reality and Visualization (ICVRV)*, 2017.

[35] Y. Kim and J.-M. Park, "Talon metaphor: Grasp and release method for virtual reality," *Springer International Publishing AG, part of Springer Nature*, 2018.

[36] A. A. Y. S.-A. R. Lorraine Lin, Aline Normoyle, "The effect of hand size and interaction modality on the virtual hand illusion," *IEEE Conference on Virtual Reality and 3D User Interfaces*, 2019.

[37] X. X. Y. Z.-Y. L. Y. W. S.-C. Z. Hangxin Liu, Zhenliang Zhang, "High-fidelity grasping in virtual reality using a glove-based system," *2019 International Conference on Robotics and Automation (ICRA)Palais des congres de Montreal, Montreal, Canada, May 20-24*, 2019.

[38] M. Prachyabrued and C. W. Borst, "Visual feedback for virtual grasping," *IEEE Symposium on 3D User Interfaces*, 2014.

[39] J. S. Thomas Ullmann, "Intuitive virtual grasping for non haptic environments.," *2Computer Graphics and Applications*, 2000.

[40] D. L. D. X. H. A. Yang, Y.; Weng, "An improved method of pose estimation for lighthouse base station extension.," *Sensors 2017, 17, 2411*, 2017.

[41] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 4, pp. 451–462, 2000.

[42] S. Hochreiter and Schmidhuber, "Long short-term memory," *Neural computation,9(8):1735–1780*, 1997.

[43] H. W. D.-Y. Y. W.-k. W. W.-c. W. Xingjian SHI, Zhourong Chen, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in Neural Information Processing Systems 28*, 2015.

[44] J. R. Lewis, "Ibm computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use," *Int. J. Hum.-Comput. Interact., vol. 7, no. 1*, 1995.

[45] S. A. M. Usoh, E. Catena and M. Slater, "Using presence questionnaires in reality," *Presence: Teleoper. Virtual Environ., vol. 9, no. 5*, 2000.

Appendix A

The paper **L. Almeida, E. Lopes, B. Yalcinkaya, R. Martins, A. Lopes, P. Menezes, and G. Pires (2019).** Towards natural interaction in immersive reality with a cyber-glove was presented at the Conference: 2019 IEEE International Conference on Systems, Man, and Cybernetics.

Towards natural interaction in immersive reality with a cyber-glove *

L. Almeida[1,2], E. Lopes[1], B. Yalçinkaya[1], R. Martins[1], A. Lopes[1], P. Menezes[2], and G. Pires[1]

Abstract— **Over the past few years, virtual and mixed reality systems have evolved significantly yielding high immersive experiences. Most of the metaphors used for interaction with the virtual environment do not provide the same meaningful feedback, to which the users are used to in the real world. This paper proposes a cyber-glove to improve the immersive sensation and the degree of embodiment in virtual and mixed reality interaction tasks. In particular, we are proposing a cyber-glove system that tracks wrist movements, hand orientation and finger movements. It provides a decoupled position of the wrist and hand, which can contribute to a better embodiment in interaction and manipulation tasks. Additionally, the detection of curvature of the fingers aims to improve the proprioceptive perception of the grasping/releasing gestures more consistent to visual feedback. The cyber-glove system is being developed for VR applications related to real estate promotion, where users have to go through divisions of the house and interact with objects and furniture. This work aims to assess if glove-based systems can contribute to a higher sense of immersion, embodiment and usability when compared to standard VR hand controller devices (typically button-based). Twenty-two participants tested the cyber-glove system against the HTC Vive controller in a 3D manipulation task, specifically the opening of a virtual door. Metric results showed that 83% of the users performed faster door pushes, and described shorter paths with their hands wearing the cyber-glove. Subjective results showed that all participants rated the cyber-glove based interactions as equally or more natural, and 90% of users experienced an equal or a significant increase in the sense of embodiment.**

I. INTRODUCTION

Virtual and immersive reality (VR) are technologies that can find many applications that go far beyond gaming, in areas such as rehabilitation, real estate promotion, education and medical training, museum exhibitions, showrooms, simulation of accident scenarios, police training, social training, etc. The potential of adapting any space to a dynamic new virtual world in which the user can move in, opens a whole of new challenges related to immersion. The effectiveness of VR environments strongly depends on providing the same stimuli as those experienced in the real world. In order to enhance user's immersion, embodiment and presence [1][2][3], we need to support a natural consistency between the vestibular and proprioceptive feedback in addition to the visual feedback, while enabling a precise tracking of body parts [4]. Interaction based on active movements contributes for the "sense of agency", that is, the sense of having "*global motor control, including the subjective experience of action, intention, control, motor selection and the conscious experience of will*" [5].

There are several low-cost commercial VR systems available that provide an effective user experience in large spaces (e.g., HTC Vive, Oculus Rift S), with a very reliable body tracking. For user interaction with VR, most systems use hand-based controllers with click buttons and inertial sensors. However, the interaction is not always perceived as natural, because while users hold these controllers they cannot grab or touch real objects in a mixed reality interaction, or it compromises the embodiment in virtual reality interaction. In a previous paper [6] we explored the notion of telepresence and physical embodiment. The aim was to virtually transfer the operator to the remote robot to improve teleoperation, maximizing task performance and minimizing the operator's physical and cognitive workload. One of the system's limitation was the lack of hand interaction with objects. Although human body parts were mapped through a skeleton representation, the hands and fingers were not tracked compromising the manipulation tasks. Recent approaches based on glove systems can help achieve a more natural user interaction, freeing user's hands, allowing the detection of finger movements, haptic feedback and gesture recognition [7][8][9][10]. Cyber-gloves open a new range of applications in gaming, industry, surgery training, rehabilitation and education. Gloves with haptic feedback are being proposed for hand and finger rehabilitation [8] and [11], or surgery training [12]. These glove–based systems aim to provide feedback to the users to enable the perception of virtual objects. Several technological approaches to this problem have been proposed in literature, which include: the use of force sensitive resistors combined with vibro-tactile actuators to provide force feedback to the user [8]; fingertip contact pressure sensors, capable of providing vibratory and visual stimulation [11]; or twisted string actuation integrating force sensors and small-size DC motors [13]. The recognition of human-hand postures in real time is being addressed in [9], which proposes a monochrome glove patterned with Augmented Reality (AR) using a camera to track each marked finger and the palm of the hand. The use of bend sensors and IMU (inertial measuring unit) is also commonly used to track, respectively, fingers and hand position and orientation [11]. The use of cyber-gloves for gaming is proposed in [14] and [15]. The work in [15] describes an exoskeletal VR glove that tracks the user's physical finger movement, and is capable of translating the movement to

*This work was supported by the project HTPDIR - Human Tracking and Perception in Dynamic Immersive Rooms, POCI-01-0247-FEDER-017644, {SketchPixel, IPT, UC} and developed at Lab. VITA.IPT - Life Assisted by Intelligent Environments, Tomar, Portugal

[1]Polytechnic Institute of Tomar, 2300 Tomar, Portugal. {laa, anacris, gppires}@ipt.pt, {elioclopes, berilyalcinkayaa, vanheelsing.martins}@gmail.com

[2]Institute of Systems and Robotics, University of Coimbra, 3000 Coimbra, Portugal. paulo@isr.uc.pt

virtual fingers in a game environment. Haptic feedback is provided by attaching motors to each finger joint. A VR glove for falconry is proposed in [14], which is intended to give the player the illusory sensation of a falcon standing in their hand. All these approaches are prototypes and do not yet have the desired usability and wearability combined with reliable exteroceptive perception, which makes the user's interaction still not very natural.

In this paper we propose a cyber-glove to improve the immersive sensation and the degree of embodiment in virtual and mixed reality interaction tasks. In particular, we are proposing a glove system that can track wrist movements, hand orientation and finger movements. Most VR systems do not provide a decoupled position of the wrist and hand, although actual hand manipulation tasks require that these two movements are independent. Thus, the perception of these two different degrees of freedom significantly increases the embodiment perception in all kind of hand interaction tasks, for which the joint between the hand and the wrist is not rigid. Furthermore, the curvature angle of the fingers is detected, supporting an effective perception of finger movements. As additional features, the glove has a heart rate sensor and a range sensor on the back of the hand that detects near obstacles, alerting the user with a vibro-tactile stimulation. The system is being tested in a virtual environment where users have to perform interaction tasks such as grab and rotate door handles, push or pull drawers. We compare the naturalness, usability, immersion and embodiment perception using the developed cyber-glove and the HTC hand trackers. These interaction tasks are part of a global set commonly used by a person when exploring a real house. This work is being developed in the scope of a major project (called HTPDIR) in partnership with a company that aims to do real estate promotion. The tests aim to understand how gloves can improve user interaction with the house being visited.

II. The System – Immersive room testbed

The present application builds a VR immersive scenario based on Unity 3D providing egocentric visualization and interaction. It is being developed targeting mixed reality applications where real objects are dynamically mapped in the VR scenario through several Kinect RGB-D sensors [16]. This paper is only focused on the design and development of an immersive cyber-glove and its use in VR environments that include manipulation tasks.

A. Immersive cyber-glove – hardware architecture

The immersive glove detects wrist, hand and finger movements with the purpose of complementing the immersive tracking system. The absolute position of the wrist is tracked with HTC Vive trackers, while the hand and finger movements are tracked with our own customized glove (see Fig. 1). The hardware architecture is shown in Fig. 2. The glove prototype is composed of the following modules: an IMU Sensor (BNO055) with ARM Cortex M0 incorporated, an wi-fi module ESP8266-07 with a Tensilica Cadence L106 embedded microcontroller, a PPG sensor (MAX30105) that is responsible for detecting the heart rate (that can be used to evaluate emotional reactions), a micro-laser range sensor (VL53L0X) responsible for measuring the distance to physical obstacles, 5 Flexible sensors (Spectra Symbol 2.2) that sense the curvature angle of the fingers, 5 force sensors (FSR 400) that provide touch information in mixed reality interaction, a vibro-motor that is actuated when the back of the glove is near a physical obstacle (for safety reasons) and a LiPo battery (BAT525). The BNO055 is a system in a package, that integrates a triaxial accelerometer, a triaxial gyroscope, a triaxial geomagnetic sensor and a 32-bit microcontroller. It merges the accelerometer, gyroscope and the magnetometer within 9 degrees of freedom, returning an absolute orientation with a high throughput without distortion of the magnetic field. The microcontroller of the ESP8266-07 module receives via I2C the data from BNO055, namely the rotation in quaternions and the acceleration data in x, y, z coordinates. The ESP8266-07 also interfaces all the remaining analog, digital and I2C sensors, computes the heart rate, and sends to BNO055 its initial configuration settings. The wi-fi module of the ESP8266 sends all glove's data in UDP/IP packets to the Unity application running on the host PC that does the data processing and visual rendering. To prolong the battery autonomy of the cyber-glove, several sensors were programmed to be in "sleep mode" whenever they are not being used. In normal operating mode the cyber-glove power consumption is about 95 mA, while in "sleep mode" it is about 30 mA. The minimum duration of the battery in the normal operating mode is approximately 8 hours. The glove was designed in SolidWorks and printed in a 3D printer Sigma using FilaFlex material, a very resistant and flexible material (elongation at break – 665%) that provided a good hand fit sensation. The palm of the hand is free which is quite important for the mixed reality experiments.

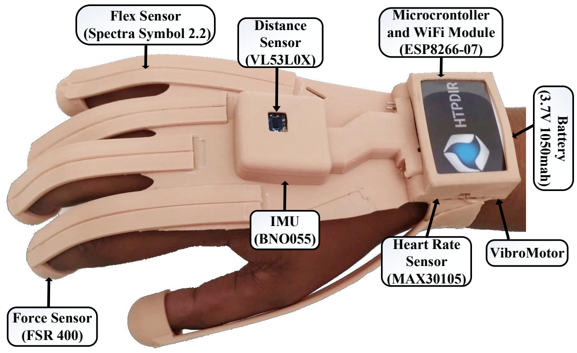


Fig. 1. Photo of the developed cyber-glove prototype (HTPDIR). Description of main hardware components integrating the cyber-glove.

B. Immersive environment - Unity 3D

Each task and effect created in the virtual environments is based on various Unity scripts and GameObjects. The script that interacts with the glove receives all the data via UDP packets every 20 ms, namely, angular position of the

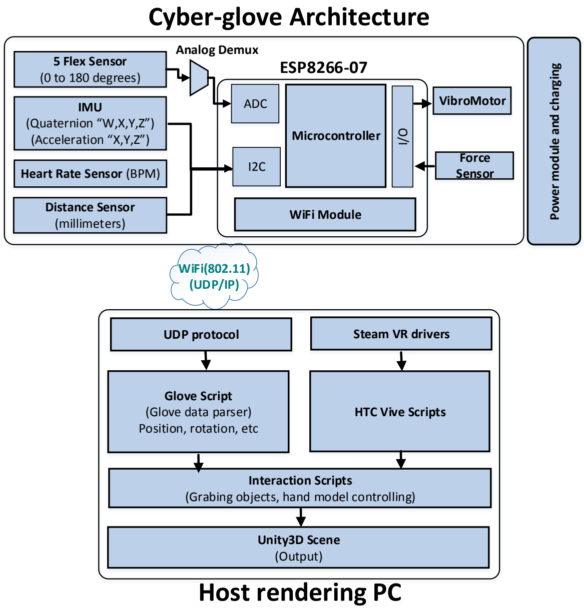


Fig. 2. Cyber-glove and 3D Unity framework architecture

hand, acceleration, range sensor distance, heart rate, fingers curvature, pressure feedback, and sends commands to actuate the vibro-tactile motors that provides haptic feedback to the user. These data are forwarded to specific scripts attached to each object to be manipulated. The hand's script receives the quaternions to animate the hand model (see Fig. 3). This hand model is attached to the wrist position detected by the HTC Vive Tracker. The wrist reference system $\{x,y,z\}_{wrist}$ has 6-DoF to which the hand reference system $\{x,y,z\}_{hand}$ is attached, having 3-DoF (roll, pitch, yaw). The reference system $\{x,y,z\}_{hand}$ is a translation of $\{x,y,z\}_{wrist}$ along the z-axis. Additionally to 3D position we get linear acceleration values for each coordinate. Fingers curvature values are obtained from flexible sensors that return a bending angle in a range of 0^o to 180^o and feed a Unity's Hinge Joint component that couples two rigid GameObjects. This enables a rotation along one of the common axis, reproducing prehensile gestures. In the extremity of each finger there is a force feedback sensor to inform the VR application that thumb and index finger are touching each other (i.e. a gesture event) or that the fingers are in contact with a real object. Interaction with objects is managed through scripts that parents the object to the hand on pickup. The attachments between objects can be rigid or use a Joint to integrate force feedback for the physics engine. These scripts also define the pick-up and release methods. When a hand and finger models are in contact with virtual objects, the respective Collider triggers the vibrotactile motors to simulate the real touch feedback. The heart rate sensor provides information that can be used to assess the user's engagement and for example change the application dynamics.

For this experiment, a virtual door scenario was created to compare the performance of the cyber-glove and the

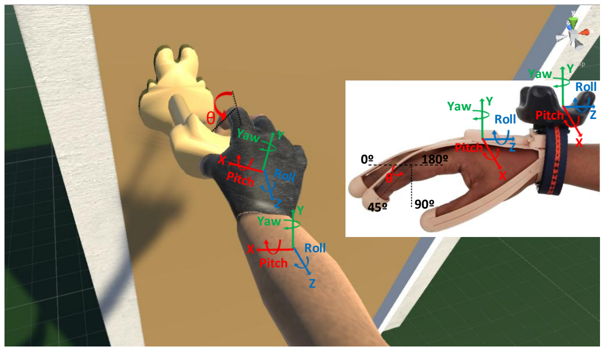


Fig. 3. Picture of the immersive scenario during a hand manipulation task: door handle rotation. Reference systems of hand and wrist in the immersive environment from an egocentric view, and respective reference systems of the cyber-glove and wrist tracker in real world.

HTC Vive Controller. Two input methods were implemented to open the door: the "DoorViveController" method that enables door opening using the HTC Vive Controller and the "DoorGlove" that refers to the door being opened by the cyber-glove. In "DoorViveController" the user presses a controller's button to grab the door's handle while in "DoorGlove" the user closes/opens his/her hand (fingers) to grab or release the door's handler. Both door-opening methods use the parenting concept to rotate the door's handle and relies on a Hinge Joint. A "Quaternion.LookRotation" method enables the rotation of the door's handle regarding to the hand position. For the "DoorGlove", hand's position results from a translation of the wrist position, provided by Vive Tracker, however their orientations can be independent (see Fig. 3. For the "DoorViveController", the hand and wrist are a rigid body with position and orientation provided by the HTC Vive controller.

The door's handle rotation is computed using the EulerAngles method and it is limited to predefined angles. Consequently, the script that rotates the door is enabled only if the door's handle rotation reaches a predefined angle. After the door unlock, the position of the hand is used to compute the door's rotation angle.This component is attached to the door frame through a Hinge Joint, which enables a rotation along a vertical axis.

In order to simplify the process of modeling real objects, we created a tool for placement of virtual points in a real scenario. To map real objects in our VR environment, we initially mark the corners of an object using the HTC Vive controllers and then, with the tool, create a virtual object anchored on those points.

III. Method

The evaluation of virtual or mixed reality applications requires an analysis of factors like naturalness, usability, immersion, embodiment and task performance, which can be assessed through user's actions such as VR 3D navigation and manipulation. This section describes the proposed evaluation methodology for this VR application and devices.

A. Participants

Twenty-two participants (7 women and 15 man) from the Polytechnic Institute of Tomar were invited to test the system. Participants were mainly students and researchers from engineering courses. Participants were aged between 20 to 46 years old (μ=26.56 , σ=6.66). Participation in the experiment was voluntary. Four of the 22 participants never had contact with video games technology and only 3 subjects had previous experience with interaction devices like the proposed cyber-glove.

B. Materials

Users viewed the virtual and the mixed reality environment through a head mount display (HMD), which is a fully immersive reality helmet that presents three-dimensional stereoscopic views. The HMD is part of the Vive VR System having two AMOLED screens, a resolution of 1080 x 1200 pixels per eye (2160 x 1200 pixels combined), a refresh rate of 90 Hz and a field of view of 110 degrees. Internal inertial sensors (accelerometers and gyroscope) and an outside-in laser tracking system provided user's head position and orientation to render the virtual world accordingly. User's hand movements and gestures were tracked either through HTC Vive controllers, or through a wrist-tracker strapped around the wrist. These trackers, when attached to a real object provided also its position. Additionally, the cyber-glove enabled the mapping of the movements of the wrist, hand and fingers in the immersive environment.

C. Experience design

One of the applications of this work is real estate promotion with virtual enviroments. A typical task commonly performed at home has been designed, *a person crosses the several divisions of a house and for this he/she has to open a door and transpose it*. One of the focus was on the embodiment and realism of the movement of the hand during the handle rotation, for which the detection of movement of the wrist and hand are essential. This simple task comprises a series of small steps that people are used to do in the real world. However, the recreation in the virtual environment presents some technological challenges. Thus we have designed a virtual door and a metaphor to transpose it. Each person, wearing a VR HMD system that provides an egocentric view, was invited to perform the following door opening based sub-tasks (see Fig. 4):

- A - Walk to the door
- B - Unlock door (rotation of handle)
- C - Push the door
- D - Pass through the door

These tasks were repeated by each subject 2 times:
1) holding the standard HTC Vive hand controller that has a click button interaction, and provides position and orientation. To grab the door's handle user had to press a button; and
2) wearing the developed cyber-glove, that frees the hand and enables grab functionalities. To grab/release the door's handle, the user had to close/open the hand (fingers).

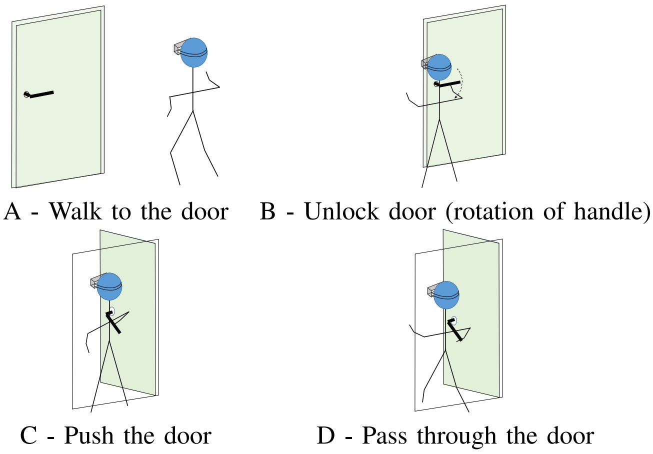


Fig. 4. Steps to open the door in the immersive environment.

Furthermore, qualitative and quantitative evaluations were performed, using the two hand-based input devices. Figure 5 shows a photo of a subject using the the cyber-glove while performing the immersive interaction task.

D. Experiment procedure

Subjects were invited individually to the lab and informed about the procedure to open the virtual door. One of the project's researcher was in charge of helping the subjects to wear the equipment, and a software application recorded the performance measures during task execution. The experiments involving the HTC Vive controller and the cyber-glove were performed randomly and at the end, users filled subjective questionnaires.

E. Evaluation metrics

Qualitative and quantitative measures were accessed for the two different hand-based input devices:

1) Efficiency measures:

- Time - measures the time taken by a person to open a door and pass through it:
  - Total time: overall time to accomplish the task;
  - Sub-task time: measure the time taken by a person in each door opening tasks (A, B, C and D);
- Length - length of the path described by the hand in each sub-tasks.

Fig. 5. Photo taken at a national exhibition of a subject using the cyber-glove while performing the immersive interaction task.

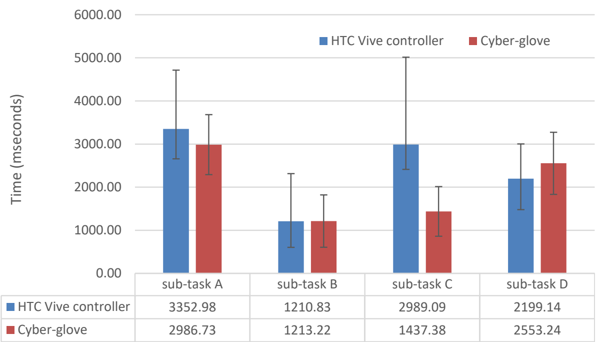


Fig. 6. Mean task-time performance of participants for each sub-task of the virtual door opening global task (HTC Vive controller vs cyber-glove).

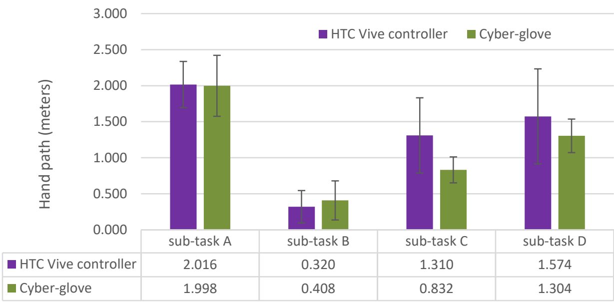
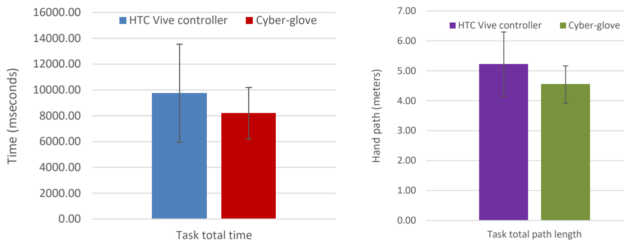


Fig. 7. Mean length of the path described by the hand of participants, for each sub-task of the virtual door opening global task (HTC Vive controller vs cyber-glove).



a) b)

Fig. 8. a) Total mean task-time, and b) total mean length of the path described by the hand of participants while performing the virtual door opening global task, HTC Vive controller vs cyber-glove.

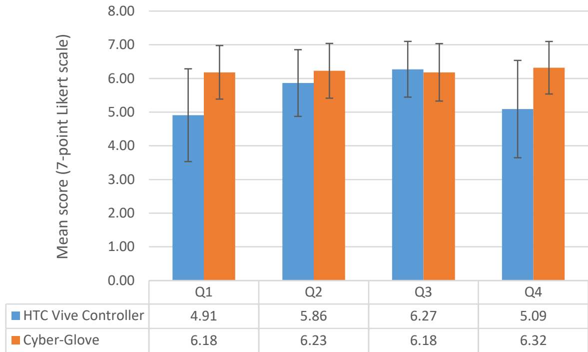


Fig. 9. Results of subjective questionnaires to participants to evaluate the virtual door opening task.

2) Immersion and presence: In order to evaluate the experience qualitatively, the subjects were invited to fill a questionnaire on a scale from 1 to 7, where 1 meant *very weakly* and 7 *very strongly*, enabling to determine issues like naturalness (question Q1), usability (question Q2), immersion feeling (question Q3) and sense of embodiment (question Q4). These questions were adapted from IBM Computer Usability Satisfaction Questionnaire [17] and from Usoh and Slater Presence Questionnaire [18].

- Q1 - How natural was the interaction with the VR environment?
- Q2 - How easy was manipulating and moving objects in the VR environment?
- Q3 - How strongly was the immersion feeling in the VR environment?
- Q4 - Did I feel that my own hand was manipulating and moving objects in the VR environment?

IV. Results

A. Task Performance

1) Objective Results: Figure 6 shows the mean of task-time performances in each sub-tasks of the virtual door opening task, using the HTC Vive controller or the cyber-glove. Statistical significance was assessed using ANOVA (analysis of variance) test (asterisk mark indicates statistically significant). Results show that that in sub-task C users pushed the door faster wearing the cyber-glove ($p < 0.05$):

- sub-task A: F(1,22)=0.69, p=0.4156
- sub-task B: F(1,22)=4.34E-05 , p=0.9948
- sub-task C: F(1,22)=6.51, p=0.0181*
- sub-task D: F(1,22)=1.29, p=0.2680
- Total task-time : F(1,22)=1.59, p=0.2204

Figure 7 shows the mean length of the path described by the hand in each sub-tasks of the virtual door opening task, using the HTC Vive controller or the cyber-glove. Statistical significance was assessed using ANOVA test:

- sub-task A: F(1,22)=0.014, p=0.9054
- sub-task B: F(1,22)=0.751, p=0.3952
- sub-task C: F(1,22)=9.012, p=0.0065*
- sub-task D: F(1,22)=1.781, p=0.1956
- Total task-length: F(1,22)=3.575, p=0.0718

Metric results revealed that 83% of the users performed faster door pushes, and described shorter paths with their hands wearing the cyber-glove ($p < 0.05$).

In Fig. 8 we present the total mean task-time results, and the total mean length of the path described by the hand of participants. The time and length of the global task are smaller for the cyber-glove. The statistical significance test for the total path length presents a p=0.0718.

2) Subjective Results: Figure 9 illustrates the results of the questionnaire.

The ANOVA one-way test for each question shows its statistic significance (asterisk marks):

- Q1: F(1,42)=14.10, p=0.00052*
- Q2: F(1,42)=1.77, p= 0.19018

- Q3: F(1,42)=0.13, p=0.72144
- Q4: F(1,42)=12.29, p=0.00109*

100% of the participants rated the cyber-glove based interactions as equally or more natural, and 90% of users experienced an equal or a significant increase in the sense of embodiment

B. Discussion

Objective results related with task-time performance in sub-task A ("Walk to the door") shows that there is not a significant time difference at reaching the door, neither there is any differences in the length of path described by the user's hand, i.e. the hand device does not influence this sub-task. Concerning the time performance in sub-task B ("Unlock door (rotation of handle)") the participants took the same time to perform this sub-task with both devices. Users with the HTC controller described a path slightly shorter than with cyber-glove while rotating the door's handle. According to our expectations and in relation to sub-task C ("Push the door"), users pushed the door to an angle of 60º faster with the cyber-glove than with the HTC controller. The length of the trajectory described by the hand while wearing the cyber-glove was also shorter. Users reported that releasing the door's handler was easier with the cyber-glove because they just had to open the hand/fingers, and they were not concerned about the instant to release the HTC controller button. Thus, users performed sub-task C better with the cyber-glove, being both task-time and hand's length path statistically significant. Concerning sub-task D ("Pass through the door"), users were faster transposing the door's frame holding the HTC Vive controller than when they were using the cyber-glove, however they described a longer path holding the HTC Vive controller. For this sub-task, the start matches the moment when the user removes their hand from the door handler and moves himself to a certain distance to the door,

Qualitative evaluation based on questionnaires to the users shows that the cyber-glove contributes for a greater naturalness. Factors like usability and immersion feeling seem similar for both devices, and the sense of embodiment is significantly improved with the cyber-glove. Several participants reported that with the cyber-glove they had more freedom of movement, while with the HTC Vive controller they felt hand movement constraints, and were afraid of dropping the controller during the release of the door handle. Overall, the results of the ongoing cyber-glove prototype are very promising. Yet, some limitations were identified during the experiments, namely, the size of the glove is not suitable for every users due to different hand sizes, and the movement of the virtual fingers exhibits a small latency in relation to the real fingers movement, due the animation model

V. CONCLUSIONS AND FUTURE WORK

This paper describes a cyber-glove system that tracks wrist movements, hand orientation and finger movements, aiming to improve the immersive sensation and embodiment in virtual and mixed reality environments. The proposed system is capable of decoupling the position of the wrist and hand, contributing for the sense of embodiment in 3D VR manipulation. The comparative study between the cyber-glove and the HTC Vive controller showed that the task-time performance of pushing a virtual door's, is faster when wearing the cyber-glove, and additionally the hand of the users describes shorter paths. The subjective questionnaires show that the cyber-glove contributes for a greater naturalness, present similar degrees of usability and immersion feeling, but improves significantly the sense of embodiment. Future work includes glove refining to better adapt to the different hand sizes, and improve VR models of the fingers. More complex tasks in virtual and mixed reality will be carried out to validate the overall system.

REFERENCES

[1] C. J. Bohil, B. Alicea, and F. A. Biocca, "Virtual reality in neuroscience research and therapy," *Nat Rev Neurosci*, vol. 12, no. 12, Dec. 2011.

[2] J. Steuer, "Defining virtual reality: Dimensions determining telepresence," *J. of Comm.*, vol. 42, 1992.

[3] T. B. Sheridan, "Musings on telepresence and virtual presence," *Presence*, vol. 1, no. 1, 1992.

[4] M. Slater, B. Lotto, M. M. Arnold, and M. V. Sanchez-Vives, "How we experience immersive virtual environments: the concept of presence and its measurement," *Anuario de Psicologia*, vol. 40, 2009.

[5] O. Blanke and T. Metzinger, "Full-body illusions and minimal phenomenal selfhood," *Trends in Cognitive Sciences*, vol. 13, no. 1, 2009.

[6] L. Almeida, B. Patrão, P. Menezes, and J. Dias, "Be the robot: Human embodiment in tele-operation driving tasks," in *The 23rd IEEE Int. Symp. on Robot and Human Interactive Communication*, Aug 2014.

[7] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 4, July 2008.

[8] M. Hoda, B. Hafidh, and A. El Saddik, "Haptic glove for finger rehabilitation," in *2015 IEEE Int. Conf. on Multimedia Expo Workshops (ICMEW)*, June 2015.

[9] H. Ishiyama and S. Kurabayashi, "Monochrome glove: A robust real-time hand gesture recognition method by using a fabric glove with design of structured markers," in *2016 IEEE Virtual Reality (VR)*, March 2016.

[10] P. Menezes, N. Gouveia, and B. Patrão, *Touching Is Believing - Adding Real Objects to Virtual Reality*, 01 2018.

[11] R. Tavares, P. J. Sousa, P. Abreu, and M. T. Restivo, "Virtual environment for instrumented glove," in *2016 13th Int. Conf. on Remote Engineering and Virtual Instrumentation (REV)*, Feb 2016.

[12] Q. Cao, C. Gao, Y. Wang, and T. Li, "The motion capture data glove device for virtual surgery," in *2016 IEEE Int. Nanoelectronics Conf. (INEC)*, May 2016.

[13] M. Hosseini, A. Sengul, Y. Pane, J. De Schutter, and H. Bruyninckx, "Haptic perception of virtual spring stiffness using exoten-glove," in *2018 11th Int. Conf. on Human System Interaction (HSI)*, July 2018.

[14] T. Al-Hathal and N. Fetais, "Virtual reality glove for falconry," in *2018 Int. Conf. on Computer and Applications (ICCA)*, Aug 2018.

[15] Q. Irfan, C. Jensen, Z. Ni, and S. Hietpas, "Building an exoskeleton glove on virtual reality platform," in *2018 IEEE Int. Conf. on Electro/Information Technology (EIT)*, May 2018.

[16] P. Girão, J. Paulo, L. Garrote, and P. Peixoto, "Real-time multi-view grid map-based spatial representation for mixed reality applications," in *Augmented Reality, Virtual Reality, and Computer Graphics*. Springer Int. Pub., 2018.

[17] J. R. Lewis, "IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use," *Int. J. Hum.-Comput. Interact.*, vol. 7, no. 1, Jan. 1995.

[18] M. Usoh, E. Catena, S. Arman, and M. Slater, "Using presence questionnaires in reality," *Presence: Teleoper. Virtual Environ.*, vol. 9, no. 5, Oct. 2000.