MINISTERY OF HIGHER EDUCATION AND SCIENTIFIC RESERCH

DR. YAHIA FARES UNIVERSITY OF MEDEA

FACULTY OF SCIENCES

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE



MASTER THESIS

PRESENTED IN ORDER TO OBTAIN

MASTER DEGREE IN COMPUTER SCIENCE

SPECIALTY: SYSTEM ENGENIREEING AND WEB TECHNOLOGY

PREPARED AND PRESENTED BY:

MR. BELLARBI AHMED

MR. BERRABAH ANES

THEME

GESTURE RECOGNITION BASED ON DEEP LEARNING: APPLICATION TO 3D INTERACTION IN COLLABORATIVE MIXED REALITY

SUPERVISER:

DR. TIBERKAK ALLAL

EXAMENING BOARD:

MR. MAZARI REDHA

DR. KHELDOUN AHMED

Academic Year: 2019/2020

ACKNOWLEDGEMENTS

My initial and foremost thanks go to Allah for endowing me with health, patience, and knowledge to complete this thesis.

Second, I would like to thank my supervisor Dr.Tiberkak Allal for the patient guidance, encouragement and advice they have provided throughout my time as their student.

Further, I am deeply indebted and grateful to my brother and Uncle Dr.Abdelkader Bellarbi, for the support, encouragement and facilities he has provided me. Thank you for everything.

Finally, I reserve my most special appreciation to my wonderful family. Firstly, to my parents who have given me love and unconditional support during life whenever I was in success or loss, and also for their patience, encouragement and prayers. I hope everything I have done can make you both happy and healthy. Secondly, to my two wonderful brothers as well as lovely sister, for their love, encouragement, emotional and moral support.

Ahmed Bellarbi

I would like to thank my supervisors Dr.Tiberkak Allal and Dr.Bellarbi Abd El-Kader, for the patient guidance, encouragement and advice they have provided throughout my time as their student. I have been extremely lucky to have these gentlemen who cared so much about our work, and who responded to our questions and queries so promptly.

I would also like to thank the entire people whom I encountered throughout my experience in the University of Yahia Fares, from the staff to the professors and fellow students and a special thanks to my partner whom helped to make this possible Mr Bellarbi Ahmed

Finally, my deep and sincere gratitude to my family for their continuous and unparalleled love, help and support. I am grateful to my brother and sisters for always being there for me as a friend. I am forever indebted to my parents for giving me the opportunities and experiences that have made me who I am. This one's for you. *Anes Berrabah*

Abstract

In virtual reality (VR) and augmented reality (AR), the word interaction is used to designate a set of rules and techniques that allow the user to perform interaction tasks in a virtual environment. The use of hand gestures is the most natural way to perform an interaction, for this reason gestures recognition has a part in the world of interaction.

In this context, our subject aims to develop a 3D interaction collaborative application by hand gestures in virtual or augmented reality, using gesture recognition by neural network.

To achieve our work, we have used many software tools such as visual studio and Unity 3D for the environment and Vuforia SDK for augmented reality, So we used these tools to design a virtual environment and represent virtual hands that reproduce user's hands gestures.

In the purpose to allow the user to interact in a virtual or augmented environment we have implemented a 3D interaction techniques which is "simple virtual hand" for the three main tasks of interaction: navigation, selection and manipulation. Also we have implemented a network manager to allow users to collaborate with each other.

This memory contains a state of the art that expose the necessary concepts for understanding our work, and a detailed explanation of the implementation leading to results and a presentation of all the tools used.

Key words: *Virtual reality, augmented reality, 3D interaction, gestures recognition, collaboration, co-manipulation.*

Résumé

En réalité virtuelle (VR) et en réalité augmentée (RA), le mot interaction est utilisé pour désigner un ensemble de règles et de techniques permettant à l'utilisateur d'effectuer des tâches d'interaction dans un environnement virtuel. L'utilisation de gestes de la main est la manière la plus naturelle pour réaliser une interaction, c'est pourquoi la reconnaissance des gestes a un part dans le monde de l'interaction.

Dans ce contexte, notre sujet vise à développer une application collaborative d'interaction 3D par gestes de la main en réalité virtuelle ou augmentée, utilisant la reconnaissance gestuelle par réseau de neurones.

Pour atteindre notre travail, nous avons utilisé de nombreux outils logiciels tels que Visual Studio et Unity 3D pour l'environnement et Vuforia SDK pour la réalité augmentée. Nous avons donc utilisé ces outils pour concevoir un environnement virtuel et représenter des mains virtuelles qui reproduisent les gestes des mains de l'utilisateur.

Dans le but de permettre à l'utilisateur d'interagir dans un environnement virtuel ou augmenté, nous avons implémenté une technique d'interaction 3D qui est «main virtuelle simple» pour les trois tâches principales d'interaction (navigation, sélection et manipulation). Nous avons également mis en place un gestionnaire de réseau pour permettre aux utilisateurs de collaborer les uns avec les autres.

Ce mémoire contient un état de l'art qui expose les concepts nécessaires à la compréhension de notre travail, et une explication détaillée de la mise en œuvre aboutissant à des résultats et une présentation de tous les outils utilisés.

ملخص

في الواقع الافتراضي و الواقع المعزز، يتم استخدام مفهوم التآثر لتعيين مجموعة من القواعد والتقنيات التي تسمح للمستخدم بأداء مهام التآثر في بيئة افتراضية أو معززة. و الطريقة الطبيعية للتآثر هي استخدام إشارات اليد، فبذلك يرتبط مجال التآثر بمجال التعرف على الإيماءاتss.

في هذا السياق، يهدف موضوعنا لنهاية الدراسة إلى تحقيق تطبيق تآثر ذات ثلاثة أبعاد بواسطة حركات الأيدي في الواقع الافتراضي أو المعزز، وذلك باستخدام التعرف على الإيماءات بواسطة الشبكة العصبية .

لتحقيق عملنا، لقد استخدمنا العديد من الأدوات البرمجية مثل visual studio و Unity 3D من اجل البيئة . و SDK Vuforia للوقع المعزز.

من أجل السماح للمستخدم بالتفاعل في بيئة افتراضية أو معززة ، قمنا بتنفيذ تقنية تفاعل ثلاثية الأبعاد وهي "يد افتراضية بسيطة" لمهام التفاعل الرئيسية الثلاث: التنقل والاختيار والمعالجة. لقد قمنا أيضًا بتطبيق مدير شبكة للسماح للمستخدمين بالتعاون مع بعضهم البعض.

تحتوي هذه الذاكرة على مستوى التقدم الجاري تعرض المفاهيم اللازمة لفهم عملنا ، وشرح مفصل للتنفيذ يؤدي إلى نتائج وعرض لجميع الأدوات المستخدمة.

Tabel of content

Table of tables ……..…….. *i*

Table of figures ……..……....……... *i*

Glossary ……..……..…… *iii*

General introduction …..……. *iv*

Chapter 1 3D interaction and Collaborative work in Mixed Reality (MR). 3

1.1 Introduction **.....3**

1.2 Definitions 4

1.2.1 Virtual Reality (VR) 4

1.2.2 Augmented Reality (AR) 7

1.2.3 Pose estimation and tracking in augmented reality 10

1.2.4 Mixed Reality (MR) 14

1.3 Mixed Reality applications and technologies 16

1.3.1 Education 16

1.3.2 Engineering 17

1.3.3 Entertainment 19

1.3.4 Health Care 20

1.4 3D Interaction in MR.....21

1.4.1 Definition 21

1.4.2 Interaction Tasks 22

1.4.3 Interaction techniques 22

1.4.3.1 Device-Based Interaction 23

1.4.3.2 Touch-Based Interaction 24

1.4.3.3 Mid-Air Gestures- Based Interaction 25

1.5 Collaborative Mixed Reality 26

1.6 Conclusion 27

Chapter 2 Gesture Recognition 29

2.1 Introduction 29

2.2 Definitions 29

2.2.1 Gesture 29
2.2.2 Gesture recognition 30
2.3 Gesture Recognition applications 31
2.3.1 Medicine and Assistive Technologies 31
2.3.2 Gaming (Entertainment) 33
2.3.3 Human Robot Interaction 34
2.4 Vision based gesture recognition 35
2.4.1 Detection 36
2.4.1.1 Color 37
2.4.1.2 Shape 38
2.4.1.3 Pixel values 39
2.4.1.4 3D Model 40
2.4.1.5 Motion 40
2.4.2 Tracking 41
2.4.2.1 Template Based 42
2.4.2.1.1 Correlation-based feature tracking 42
2.4.2.1.2 Contour based tracking: 43
2.4.2.2 Optimal estimation 44
2.4.2.3 Particle filtering 45
2.4.2.4 CamShift 46
2.4.3 Recognition 46
2.4.3.1 Appearance based techniques 47
2.4.3.2 3D Model Based Techniques 50
2.5 Conclusion 54
Chapter 3 Conception and analysis 57
3.1 Introduction 57
3.2 Description of the system 57
3.3 System specification 59
3.3.1 Functional Specifications 59
3.3.2 Technical specifications 59
3.4 System architecture 60
3.4.1 Neural network gesture recognition module 60
3.4.2 Mixed reality module and collaboration and data sharing module 70
3.4.2.1 Mixed reality 70

3.4.2.2 Collaboration (network): 71
3.5 Conclusion 72
Chapter 4 Implementation and tests 74
4.1 Introduction 74
4.2 Objectives of the realization 74
4.2.1 Interaction 3D 74
4.2.2 Gestures to recognize 74
4.3 Software environment 76
4.3.1 Unity 2019.3.0f6 76
4.3.2 OpenCV 79
4.3.3 Vuforia 80
4.3.4 Visual Studio (C # programming language) 81
4.4 Implementation 82
4.4.1 Test of the gesture recognition module 82
4.4.2 Augmented reality scene 84
4.4.3 Virtual reality scene 88
4.4.4 Network manager 89
4.2 Conclusion 90
General conclusion 91
References 92

Table of tables

Table 1-1 History of Virtual Reality (Authorstream 2020). 5
Table 1-2 History of Augmented Reality (Wikipedia 2020b) 9
Table 3-1 Parameters of the implemented neural network. 67
Table 4-1 interaction gestures. 75
Table 4-2 cross test for recognition rate on Windows. 83
Table 4-3 cross test for recognition rate on Android. 83
Table 4-4 the mean calculation time for every frame. 83

Table of figures

Figure 1-1 General principle of the registration process in augmented reality. 8
Figure 1-2 The steps for detecting a marker. 11
Figure 1-3 Some examples of marker libraries. a : Maxicode (Wang & Ye 2000), b : ARToolkit (Hirokazu & Billinghurst 1999), c : QRCode (Kan et al. 2009), d : Caltag (Atcheson et al. 2010). 12
Figure 1-4 Vuforia steps for detecting a marker. 13
Figure 1-5 A representation of MR in the Reality-Virtuality continuum based on (Milgram et al. 1994). 15
Figure 1-6 an illustration of the use MR in school. 17
Figure 1-7 An illustration of the use of MR in engineering. 18
Figure 1-8 an illustration of the use of MR in entertainment. 20
Figure 1-9 an illustration of the use of MR in Health Care (surgery). 21
Figure 1-10 Using the cell phones as tennis racquets in AR tennis game and mapping the position of cell phone's built-in camera to translate and rotate a 3D object. 23
Figure 1-11 Two-finger gestures for 3D object manipulation on small touch-based display. 24
Figure 1-12 Hand tracking illustration for 3D interaction. 25
Figure 2-1 Gesture recognition illustration. 30
Figure 2-2 Surgeon using Gestix to browse medical images (Cote et al. 2006). 32
Figure 2-3 Playing bowling using The Kinect device. 34
Figure 2-4 Interaction with a robotic arm. 35
Figure 2-5 Vision based hand gesture recognition techniques (Shamayleh et al. 2018). 36
Figure 2-6 Hand Gesture representation methods (Rautaray & Agrawal 2012). 47

Figure 2-7 Colored gloves technique by (Wang & Popović 2009). 48
Figure 2-8 Interaction technique based on color marker by (Bellarbi et al. 2011). 48
Figure 2-9 Flow diagram of the gesture recognition technique by Gabor filter proposed in (Huang et al. 2011). 49
Figure 2-10 The interactive table proposed by (Bellarbi et al. 2013). 50
Figure 2-11 Some of Depth camera technologies. 51
Figure 2-12 The Principle of 3Gear technique (Wang et al. 2011). 51
Figure 2-13 Example of a user who operates in a virtual environment, using the system proposed in (Messaci et al. 2015). 52
Figure 2-14 Hand gesture recognition and tracking technique proposed in (Taylor et al. 2016). a) Initialization: estimated model of gesture. b) Convergence towards the real gesture by applying an energy optimization function. c) Example of VR application. 53
Figure 2-15 Leap motion. 54
Figure 3-1 Overall presentation of the proposed system 58
Figure 3-2 Descriptive schema of the proposed system. 58
Figure 3-3 Brave descriptive schema of the gesture recognition module. 60
Figure 3-4 Descriptive schema neural network gesture recognition module. 61
Figure 3-5 HSV color space. 62
Figure 3-6 Method used for hand detection. 63
Figure 3-7 Result after apply the algorithm. 63
Figure 3-8 Image window divided into small spatial regions (“cells”). Local 1-D histograms of gradient directions or edge orientations are accumulated and concatenated to form the final histogram feature. 65
Figure 3-9 A simple neural network 67
Figure 3-10 Training and Recognition process. 68
Figure 3-11 Vuforia simple AR display. 70
Figure 3-12 Peer to Peer architecture. 71
Figure 4-1 Images of gestures used for 3D interaction. 76
Figure 4-2 Unity interface. 77
Figure 4-3 The Flow Chart of OpenCV Face Detection. 80
Figure 4-4 Vuforia development platforms. 81
Figure 4-5 Navigation mode. 84
Figure 4-6 Rotation mode. 85
Figure 4-7 Virtual reality scene 88
Figure 4-8 Network manager. 89

Glossary

AR	***Augmented Reality***
ASM	***Approximate String Matching***
CSCW	***Computer Supported Collaborative Work***
DOF	***degrees of freedom***
DTW	***Dynamic Time Warpping***
GUI	***graphic user interfaces***
HMM	***hidden Markov model***
HSV	***hue, saturation, lightness***
MAR	***Markerless augmented reality***
MR	***Mixed Reality***
MRI	***Magnetic resonance imaging***
NFT	***Natural Feature Tracking***
RGB	***red green blue***
SIFT	***Scale Invariant Feature Transform***
SURF	***Speeded up robust features***
SVM	***Support Vector Machine***
VE	***virtual environment***
VR	***Virtual Reality***

General Introduction

Virtual reality (VR) and augmented reality (AR) are a technology that brings together a set of tools allowing users to interact with a virtual environment, this by using 3D interaction techniques on the human faculties of user action and perception. Therefore 3D interaction is the driving component of VR / AR. Many interaction techniques exist, among them those based on the recognition of gestures.

In order to present our work in a clear way, we have divided manuscript into two parts:

- Part 1: which represents the state of the art of our work, this part consists of two chapters as follows:

- Chapter 1: 3D interaction and collaboration.

In this first chapter we will define the concept of virtual reality and that of augmented reality and mixed reality, as well as the fields of application of the two concepts. Also through this chapter we will present the notion of 3D interaction, the tasks and the different 3D interaction techniques, as well as a classic classification of these.

- Chapter 2: gesture recognition.

In this chapter we will discuss gesture recognition, with its different approaches and techniques, but for that, we will first define what gesture is, its taxonomy and its representation.

- Part 2: this second part contains a detailed presentation of the work carried out, through the following two chapters:

- Chapter 3: Conception and analysis

Before going to the conception of our application we will first go through a needs analysis through a functional description of the application, the technical and functional specifications. After that, we go through with the system architecture and the modules of our work.

-Chapter 4: Implementation and test

In this last chapter of the thesis, we will present the tools used to carry out our work, as well as a detailed explanation of the implementation.



Part one: State of the art



Chapter 1. 3D interaction and Collaborative work in Mixed Reality (MR).

Chapter 1 3D interaction and Collaborative work in Mixed Reality (MR).

1.1 Introduction

The idea of being able to enter and completely feel another world at some point in the future, being able to interact with knowledge and entertainment in a totally immersive state was naïve . However, thanks to the major Advancements and the popularity of technology in the past few decades, it led that mere idea into becoming a reality.

The promise of being able to be inside another world was resolved initially with what's called Virtual Reality, and later on similar technologies such as Augmented and Mixed Reality, using special equipment for the ultimate experience.

Moreover, the Human-Computer Interface is giving way to a Human-Human Interface mediated by computers, as Computers are increasingly used to enhance collaboration between people. As collaborative tools become more common. This emphasis on collaboration adds new technical challenges to the design of Human Computer Interfaces using Mixed Reality techniques in order to enhance remote and face to face collaboration, particularly 3D Computer Supported Collaborative Work (CSCW).

1.2 Definitions

These last two decades have seen a massive development in technologies especially in augmented reality, as we come across the terms of Virtual Reality(VR), Augmented Reality (AR) and Mixed Reality (MR) which most of the people can't tell apart .

1.2.1 Virtual Reality (VR)

Virtual Reality stands for the field of computing which has the objective of creating a virtual world, having one immerse into it and giving one the capability of interacting with this world, while using specific devices to simulate an environment and stimulate one by feedback in order to make the experience as real as possible (Howard 1991). VR consists of three key elements which are: Immersion, Perception and Telepresence.

- Immersion: Immersion is the suspension of disbelief (Kalawsky 1996) possibly being given by any media. There are three types of systems in Virtual
    - Non Immersive Systems, such as Desktops, which are not very sophisticated devices for Virtual Reality applications.
    - Fully Immersive Systems, which give the user the closest experience to reality through high quality graphics and performance as well as complete or major absence of unrelated stimuli.
- Perception: Perception is the awareness of the surroundings through physical senses. So, in order to give one the perception of something,

sensorial stimuli must be used. Two approaches (Coyne & Heidegger 1994) are usually seen:

  - The data-oriented approach, which aims immersion through quality of data, meaning the more the data seems like reality, more immersive the experience will be. That can be seen as experiences with higher definition Medias provoke more immersion to the user.
  - The constructivist approach, which, on the other hand, uses the human capability of building a reality, thus being capable of immersing one with not so high quality technologies. Evidence for this can also be found, as there's not one human being which haven't being amazed by a book.

- Telepresence: Virtual Reality also uses the concept of Telepresence, which is the characteristic of being able to feel present somewhere different of your real location, a remote presence. It was created by Marvin Minsky in 1980 (Held et Durlach 1993] and is a term widely used by the virtual reality community for it is highly connected with the concept of immersion.

According to (Authorstream 2020), we can see the in the table below the major advances in the VR from the 1965 till present day.

Table 1-1 History of Virtual Reality (Authorstream 2020).

year	Person/ People responsible	The importance
1957-1962	Morton Heilig	Sensorama simulator
1977	Dan Sandin, Richard sayre and Thomas Defanti	Interaction through body movement

1983/1984	Myron Krueger/ William Gibson(his assistant)	First virtual environment / adobting the term 'Cyberspace'
1995	Silicon Graphics	Virtual Reality Modeling Language
2001	(Maurice Benayoun, David Nahon), Barco, and Clarté	the first PC-based cubic room: SAS Cube (SAS3)
2007	Google	Street view
2010	Palmer Luckey	Oculus Rift
2012	John Carmak	software for Game using Oculus Rift(Doom3)
2014	Sony	Playstation VR
2015	Google	Google Cardboard
2016	Amazon, Apple, Facebook and others	Developing VR-related products
2016	HTC	HTC Vive stream VR headset
2018	Oculus (F8 development conference)	the Half Dome – a headset with a 140-degree field of vision
2019	Oculus	Oculus Rift S

1.2.2 Augmented Reality (AR)

Unlike VR, in Augmented Reality, users see and interact with the real world while digital content is added to it. It allows the user to see the real world with computer generated objects superimposed upon the real world. Therefore, AR supplements reality rather than replacing it (gatech 2020), it focuses on both Virtual Reality and real-world elements together, while being interactive in real time. AR uses a person's own visual and spatial abilities for interaction with the real world.

Aligning virtual objects with the real world can be done by aligning real and virtual cameras. In order to obtain a coherent augmented world combining both the virtual and the real, we must attribute to the virtual camera the same properties (extrinsic and intrinsic) as those of the real camera. Therefore, we have to determine in real time for each image the position and orientation of the camera in the real scene. The following figure (figure 1-1) illustrates the 2D-3D registration problem.

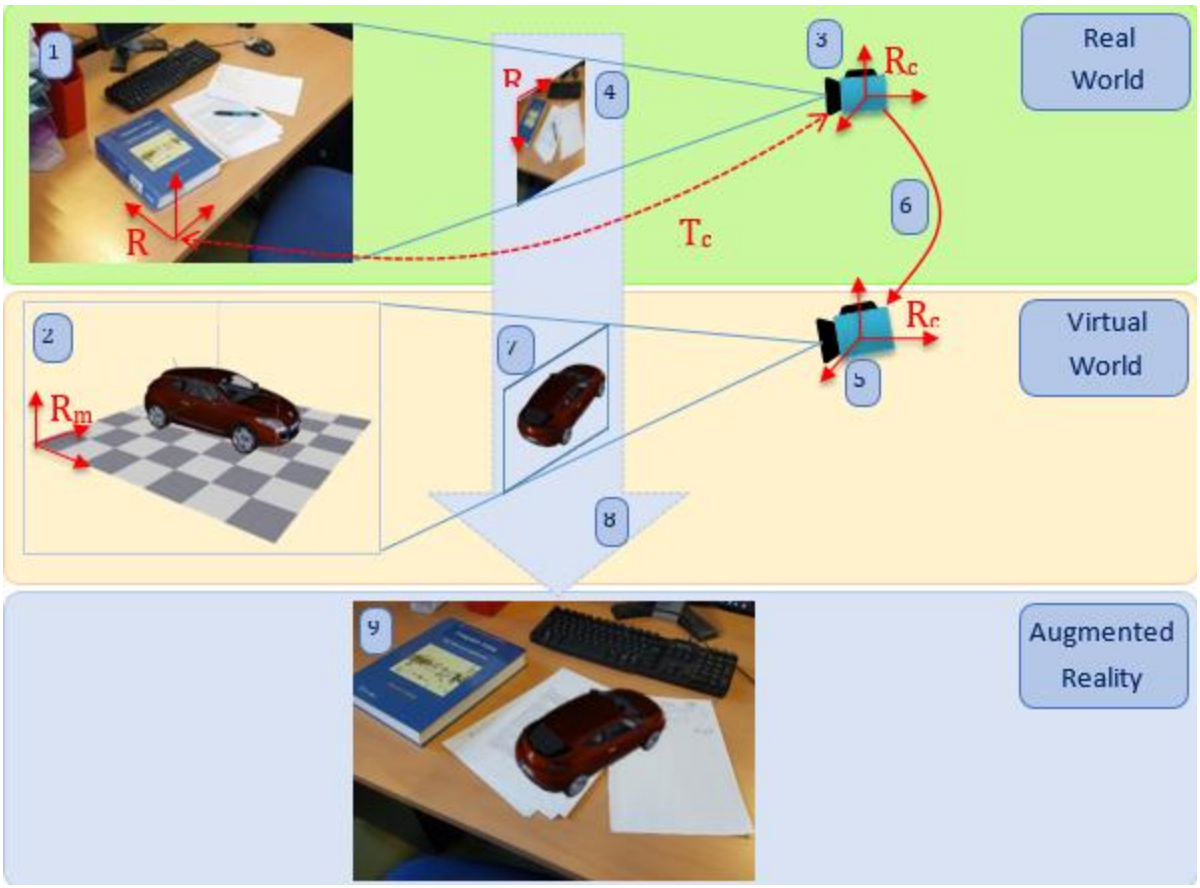


Figure 1-1 General principle of the registration process in augmented reality.

The nine elements that are mentioned above in the figure are:

1. Real environment :
2. Virtual environment.
3. Camera: which is an optical instrument used to record images?
4. Image captured: the output of the camera in a two dimensional form.
5. Virtual camera: a function of the animation software that works and behaves in the same way a camera or digital camera would in real-world situations
6. Alignment of the virtual camera with the real camera.
7. Projection space.

8. Real / virtual mixing.
9. Augmented reality.

Augmented reality is considered as a newer technology while in fact it's available for over three decades, it gradually emerged from VR and adopted the term "Augmented Reality" in 1992.

According to (Wikipedia 2020b), the table down below shows the major advances of AR.

Table 1-2 History of Augmented Reality (Wikipedia 2020b)

Year	Person/people responsible	The importance
1966	Ivan Sutherland	Head-mounted display
1975	Myron Krueger	Videoplace that allows interacting with virtual objects.
1980	Steve Mann	First Wearable computer (EyeTap)
1992	Louis Armstrong	Developing First Functioning AR system (Virtual Fixtures)
2000	Bruce H. Thomas	Developing ARQuack (First outdoor mobile game)
2000	Hirokazu Kato(Nara Institute of Science and Technology)	Creating ARToolKit
2003	NFL	Using skycam

2004	Trimble Navigation and HIT Lab	First outdoor AR mounted-Helmet
2008	Wikitude AR Travel Guide	G1 android phone
2009	Adobe Flashe	Integrating AR to web browsing (FLARToolkit)
2012	Lyteshot	AR gaming platform using smart glasses
2013	Meta	Meta 1 developer Kit
2014	Google	Releasing Google Glasses
2015	Microsoft	Windows holographic
2016	Niantic	Pokemon Go
2017	Magic Leap	Magic Leap One
2019	Microsoft	Windows hololens2

1.2.3 Pose estimation and tracking in augmented reality

In augmented reality, the 3D pose (position and rotation) of the user has to be determined to be able to add virtual objects in specified locations. Several approaches have been proposed using simple webcam images (RGB).

A. Planer marker tracking

Some approaches used predefined markers (fiducial or planer markers). The use of these type of markers were democratized with the immergence of the ARToolkit library (Hirokazu et Billinghurst 1999), most of those markers have a rectangular shape, black edges on a white background and have a code to identify them.

The marker detection algorithm goes through several steps before arriving at the desired result, mainly: acquisition, detection, spatial localization, and processing.

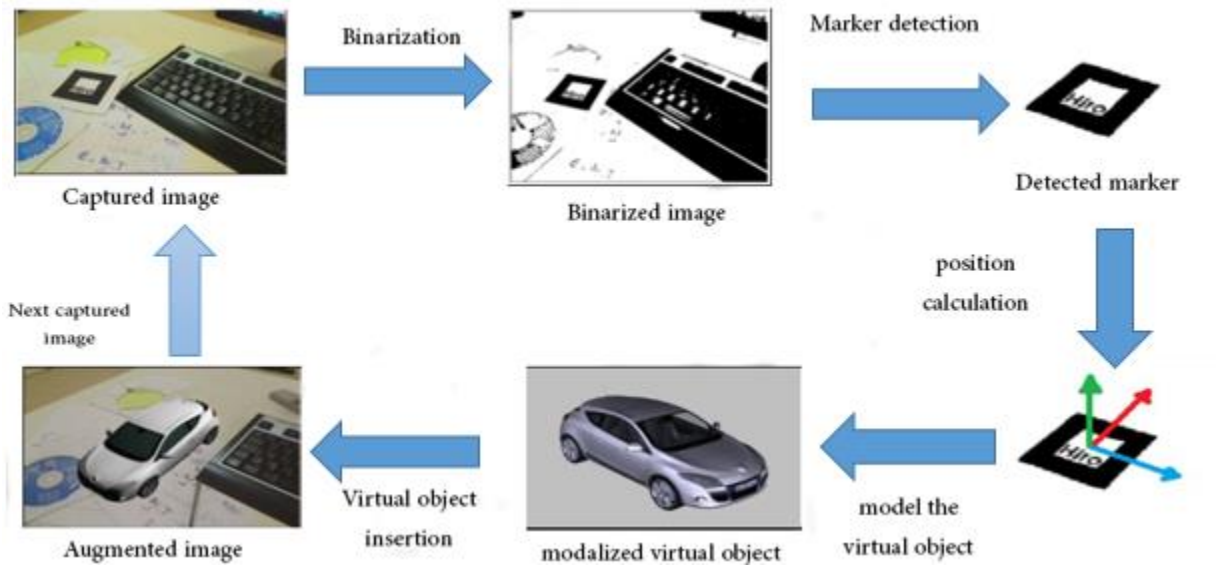


Figure 1-2 The steps for detecting a marker.

Many techniques and libraries of markers with different shapes have been proposed to ensure stable and precise registration in RA. However, the major disadvantage of markers is the heterogeneity with the scene to be augmented. Figure (figure 1-3) shows some libraries of this type of Markers.

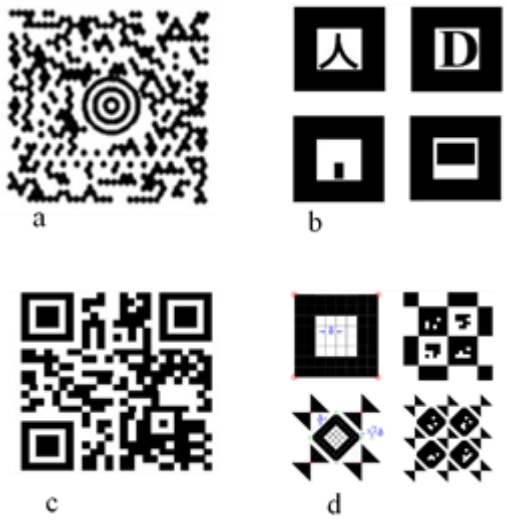


Figure 1-3 Some examples of marker libraries. a : Maxicode (Wang & Ye 2000), b : ARToolkit (Hirokazu & Billinghurst 1999), c : QRCode (Kan et al. 2009), d : Caltag (Atcheson et al. 2010).

B. Natural feature tracking

Other proposed approaches use natural features instead of predefined markers, in this case we can use a part or object (image) from the scene as a marker.

These type of approaches has been democratized with the evolution of the computer vision techniques, those techniques are usually based on feature description and tracking, for more details about this techniques we can read (Marchand et al. 2016) and (BELLARBI 2017)

Several libraries using this approach have been developed which offer stable and precise estimation of the user position, and they are easy to use. Some of those libraries have free version such as ARToolkit 5 or Vuforia.

Natural Feature Tracking (NFT) contained in the vuforia library functions to determine whether an image can be recognized or not based on pattern

recognition by detecting the angular points of the pattern in the image. The goal is to make detecting feature patterns easier. Basically the Vuforia library uses NFT (Natural Feature Tracking) in which there is a natural feature tracking approach namely SIFT (Scale Invariant Feature Transform). SIFT is used to detect "feature points" and determine the scale of the marker (target image) by mapping the coordinates value (Vuforia 2018). Figure (figure 1-4)below shows the steps to detect markers in Natural Feature Tracking (ULFAH et al.2020).

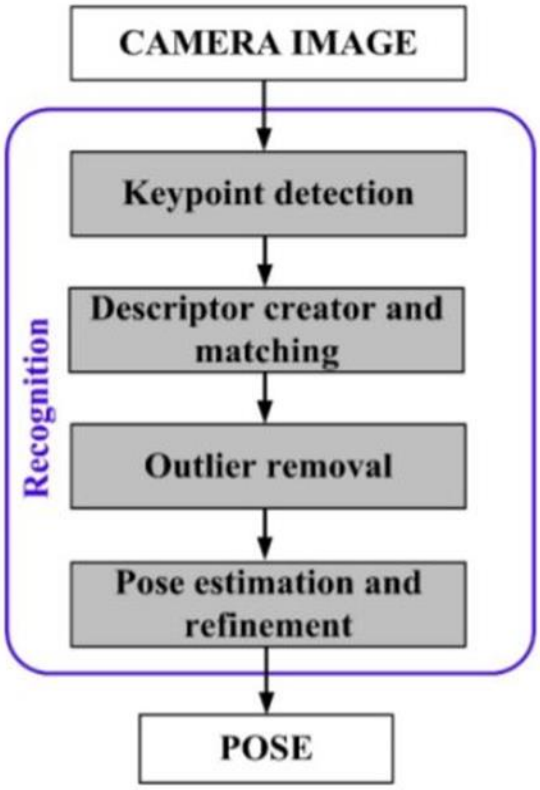


Figure 1-4 Vuforia steps for detecting a marker.

C. Markerless augmented reality

Markerless augmented reality systems integrate 3D virtual objects into a 3D real environment in real-time, enhancing user's perception of and interaction with the world. Its basic difference from marker based AR systems is the method used to place virtual objects in the real world. This approach is not based on the use of traditional artificial markers that need to be placed in the world to be tracked by the system in order to calculate their position and orientation. In MAR any part of the real environment may be used as a marker that can be tracked in order to position virtual objects. Therefore, there are no ambient intrusive markers that are not really part of the environment. Furthermore, MAR counts on specialized and robust trackers. Another advantage is the possibility of extracting from the environment characteristics information that may later be used by the MAR system. Nonetheless, tracking and registration techniques become more complex in MAR systems. Another disadvantage emerges in online MAR since it presents more restrictions (TEICHRIEB et al. 2007).

1.2.4 Mixed Reality (MR)

Mixed reality is the merging of the real and virtual worlds, producing an entire new environment and visualization. Where physical and digital objects co-exist and interact in real time. It does not exclusively take place in either the physical or virtual world, but is a hybrid of reality and virtual reality.

MR is a broad concept belongs to the Reality-Virtuality continuum (Milgram et al. 1994) that consists of Real Environment, AR, Augmented virtuality and Virtual Environment which is demonstrated in figure below (Figure 1-5)

On the reality-virtuality continuum, an environment consisting solely of real objects is one, and an environment consisting solely of virtual objects is the other extreme. All other forms of augmented environments, real or virtual, fall somewhere along the continuum, with an increasing degree of virtualisation towards the VE extreme, in which the perception of the real world is completely replaced with a simulated (or virtual) environment. A real environment would naturally include any real-world scene viewed directly by a person, but also any non-augmented real-world scene viewed from a display. Virtual environments would include, on a basic level, any completely computer-generated scenes viewed from a display, and on more complex levels also any fully computer-generated virtual systems and environments.

The basic concept of MR environment would be one in which the real world and virtual world objects are presented and interact together within a single display (or a single environment).

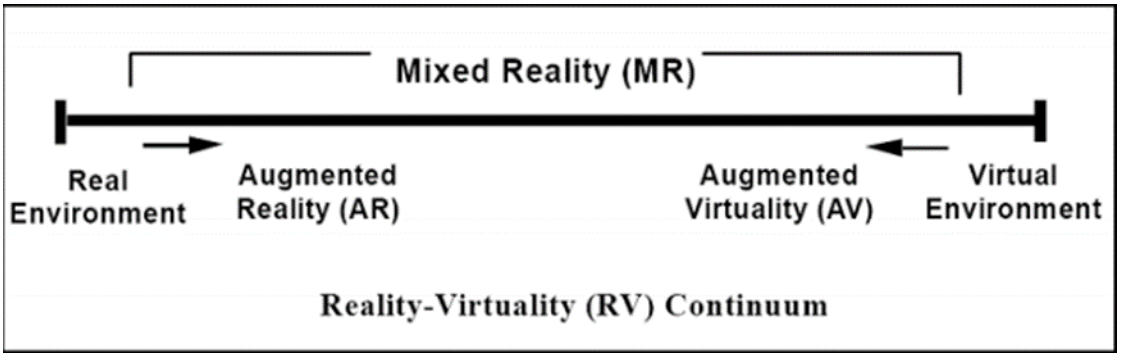


Figure 1-5 A representation of MR in the Reality-Virtuality continuum based on (Milgram et al. 1994).

1.3 Mixed Reality applications and technologies

At present, Mixed Reality (MR) systems are still being developed and fine-tuned. This technology is still not as widely available as separate VR and AR devices or software, although there are some real-world examples of mixed reality technologies in use today.

Mixed reality devices like the Microsoft HoloLens2 (Microsoft 2020) aren't found everywhere just yet, although, with a steadily increasing number of applications and growing use in certain industrial and commercial fields, they certainly could be in the not-too-distant future.

As we'll see, most of the current use cases of mixed reality technologies are bespoke to a specific task or objective and have different applications within different fields or sectors.

1.3.1 Education

Mixed reality technologies are being used within the education industry to both enhance students' ability to learn and take in information. It also gives the students the opportunity to personalize the way they learn.

Using 3D projections and simulations, students can interact with and manipulate virtual objects in order to study them in a way that is relevant to themselves and their studies. By inserting three-dimensional objects into a classroom as a means of gauging the size, shape, or other features of a defined "virtual" object, students can gain a deeper sense of understanding as to what it is they're studying especially in Geometry, Anatomy and other subjects

Some ways that MR can help in the classroom:

- Interact with the environment in an immersive experience.
- Touch and manipulate objects.
- It is an engaging and fun way of learning.
- MR can teach any kind of subject.

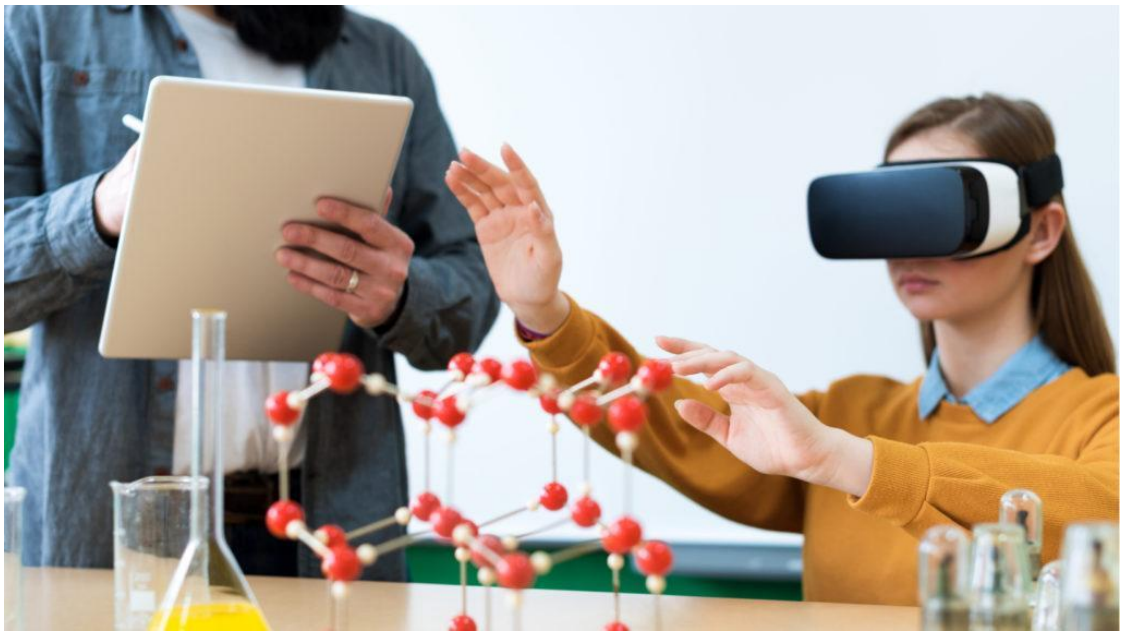


Figure 1-6 an illustration of the use MR in school.

1.3.2 Engineering

Mixed reality in engineering is slowly but surely becoming a game-changer. From 3D modeling and virtual sculpting to remote repair guidance and project monitoring apps. There are various ways in which the engineering sector has begun to take advantage of mixed reality devices.

Some benefits in Engineering:

- Real-time simulation of engineering processes.
- Use MR with an industrial IOT device to monitor services.
- Engineering training.

For example, using 3D modeling apps on mixed reality devices, professionals are able to build their projects up in a shared virtual environment. This type of detailed 3D modeling + collaboration gives engineers the best chance for spotting errors while also allowing real-time manipulation of their designs. The collaboration environment allows supervisors to evaluate and check their 3D designs in real-time.

Project monitoring apps featuring MR, are also growing in popularity due to their ability to project 3D or 4D design models over structures as they are being built. This can then help engineers and construction workers visualize progress as well as to inspect the quality of what has already been built.



Figure 1-7 An illustration of the use of MR in engineering.

1.3.3 Entertainment

VR entertainment has been around for decades and because of this, it's the most acclimatized and most widely available of the three technologies. Augmented reality games and apps are coming out thick and fast since the success of Pokémon Go (Pokemongolive 2020) and there's little reason to suspect they'll disappear out anytime soon, but the mixed reality in entertainment is not only reserved for gaming.

AR and VR technology is improving so much that the film and big screen industry are after the vision of incorporating them into their movies. The idea of interactive storytelling which is a mix between a game and movie will open up lots of new doors for mixed reality, Mixed reality entertainment is already here with companies like Magic Leap (Magicleap 2020) and Industrial Light And Magic (ilm 2020) all looking to delve into mixed reality entertainment. For example, Magic Leap's is embracing mixed reality in cinema by using what they call a Dynamic Digitised Lightfield Signal. Magic Leap's tech projects images directly into the eye, without the need for it to bounce off an object and then head towards the eyes. This tricks the brain into thinking the object is there when in reality, it is a projection.



Figure 1-8 an illustration of the use of MR in entertainment.

1.3.4 Health Care

When it comes to healthcare, mixed reality technologies have many potential applications. The most obvious is training and education. An example is the over-the-shoulder surgeries, where surgical students can be taught remotely by experts as they perform surgeries in real-time.

Another example is interactive learning. Topics like anatomy with mixed reality technology can be used to map the different layers of the human body. Being able to produce three-dimensional models of the anatomy complete with information accessible by just a simple gesture could change the way health care and medicine is taught.

MR will also transform the way in which medical students learn, using three-dimensional holograms in a virtual environment rather than two-dimensional diagrams from medical textbooks in base reality.



Figure 1-9 an illustration of the use of MR in Health Care (surgery).

1.4 3D Interaction in MR

1.4.1 Definition

In computing, 3D interaction is a form of human-machine interaction where users are able to move and perform interaction in 3D space. Both human and machine process information where the physical position of elements in the 3D space is relevant. The 3D space used for interaction can be the real physical space, a virtual space representation simulated in the computer, or a combination of both. When the real space is used for data input, humans perform actions or give commands to the machine using an input device that detects the 3D position of the human action. When it is used for data output, the simulated 3D virtual scene is projected onto the real environment through one output device or a combination of them.

1.4.2 Interaction Tasks

In the mixed environment, the interaction is essential. We can design or imagine multitudes to realize tasks for specific applications of AR or VR, however considering that base interaction tasks are practical tool that allows the realization of complex tasks in all applications.

Bowman (Bowman 1999) identified these universal tasks and classified them into four categories: Navigation, selection, manipulation and application control.

- The navigation allows the user to explore and search and maneuver in virtual space. The main components are shifting and route search.
- The selection consists of choosing explicitly one or many virtual object(s) in order to do a given action.
- The manipulation consists of modifying the attributes of the selected object (position, orientation, scale, color, texture).
- Application control includes other commands that the user can use to perform their work in the application.

1.4.3 Interaction techniques

The most focused on and probably the newest way in order to enhance Human-Computer Interaction HCI is the handheld mobile interaction.

Due to its concern, interaction in VR, AR and especially MR becomes the main and most interesting topic among researchers. One of the important

aspects is creating appropriate interaction techniques for MR applications that allow end users to interact with virtual content in an intuitive way

Few of these interaction techniques will be demonstrated below.

1.4.3.1 Device-Based Interaction

Device-based interaction technique is the technique when the device's own attributes are being utilized and users use the device as the controller for interaction purpose.

In early 2005, the device-based interaction technique was firstly implemented in handheld mobile AR interfaces as a 3D object manipulation tool (Henrysson et al. 2005a), (Henrysson et al. 2005b). For example, by using cell phones as tennis racquets in (Henrysson et al. 2005a) and using the cell phone to translate and rotate 3D objects in an augmented environment (Henrysson et al. 2005a), (Henrysson et al. 2005c) as shown in the following figure (figure 1-10).

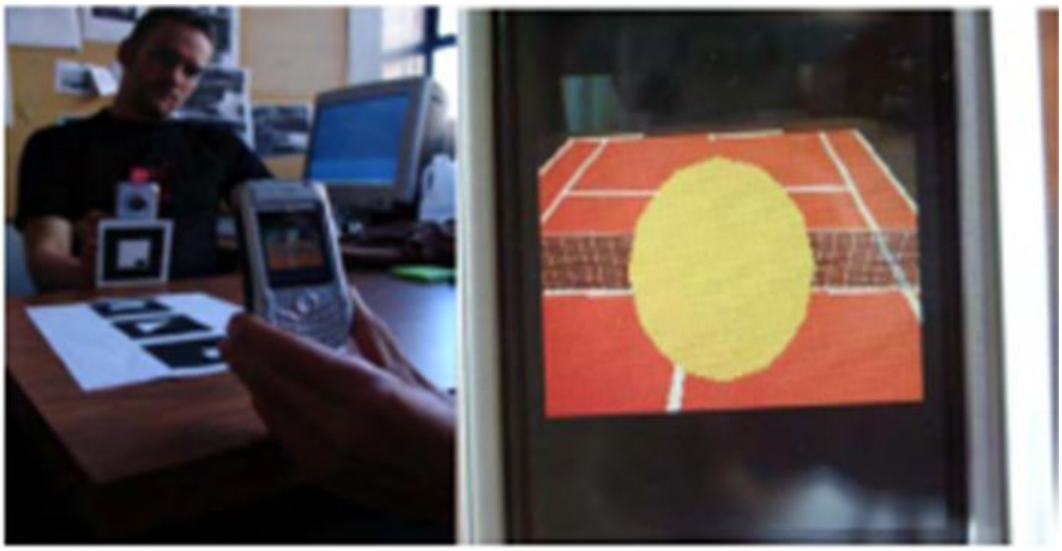


Figure 1-10 Using the cell phones as tennis racquets in AR tennis game and mapping the position of cell phone's built-in camera to translate and rotate a 3D object.

1.4.3.2 Touch-Based Interaction

Touch-based interaction becomes the main research focus recently after the introduction of touchscreen handheld mobile devices (smartphones, tablets, 2on1 laptops…) (Henrysson et al. 2005c). In the earlier approach on the mobile touchscreen, it is designed to use fingers to navigate on the screen which alters the interaction experience with mobile applications so users can directly interact with the onscreen objects through their fingers. In MR, 3D object manipulation is a challenge for interface designer since it involves the control of six degrees of freedom (6DOF) which include 3DOF for object translation (x, y, and z-axes) and 3DOF for object rotation (x, y, and z-axes), so does MR. On account that 3D object manipulation is essential in MR interfaces, touch-based interaction provides another option for users to perform 3D object manipulation.

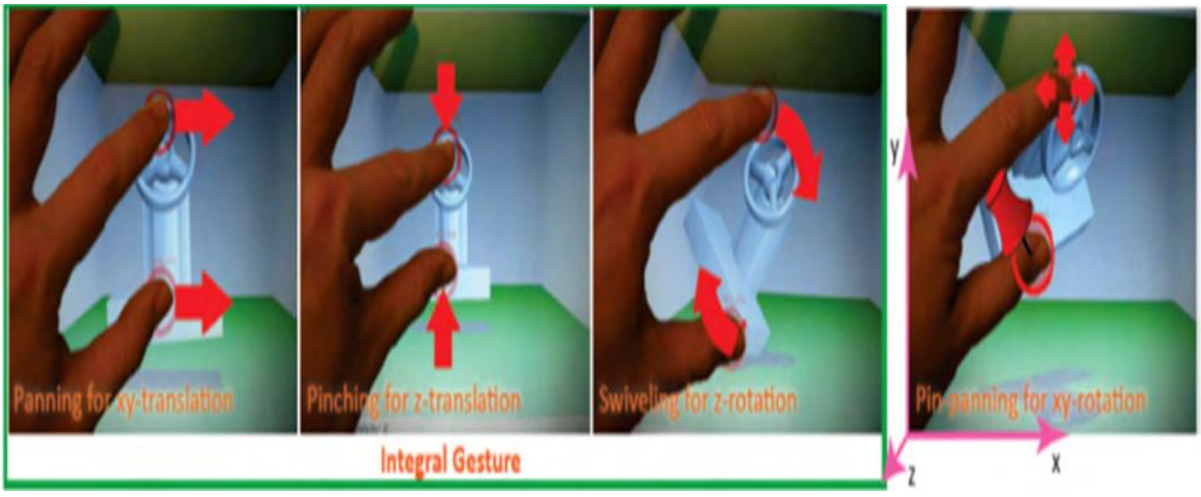


Figure 1-11 Two-finger gestures for 3D object manipulation on small touch-based display.

1.4.3.3 Mid-Air Gestures- Based Interaction

Mid-air gestures-based interaction is powerful in providing a more natural and intuitive interaction mechanism for the user. One of the earlier researches that investigates the potential to model human hands and limbs for interaction purpose was conducted by Rehg and Kanade (Rehg et Kanade 1994). In their study, human hands and limbs were modeled as articulated mechanisms and formulated as a real-time visual tracking called DigitEyes. Since their modeling approach was based on 3D kinematics, this made it possible to track any subset of hand or body states with the same basic algorithm and it was possible to track in total 27 DOF of hand model from a two-camera image sequence under orthographic projection. Which we'll see in the following chapter as Gesture recognition (Chapter 2).

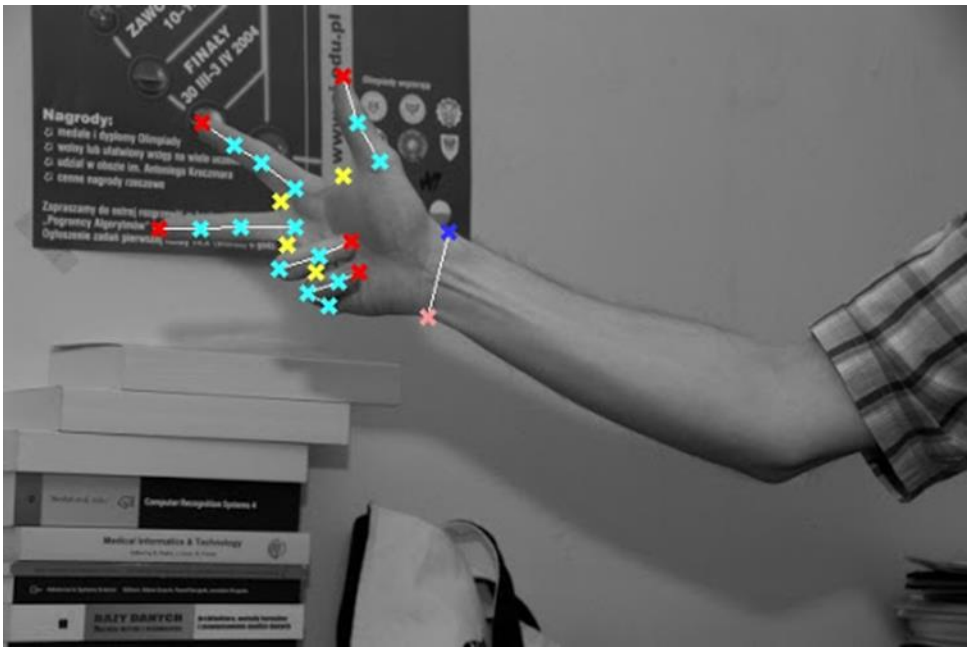


Figure 1-12 Hand tracking illustration for 3D interaction.

1.5 Collaborative Mixed Reality

Mixed Reality interfaces can overlay graphics and audio onto the real world. This allows the creation of MR interfaces that combine the advantages of both virtual environments and seamless collaboration. Information overlay may be used by remote collaborators to annotate the user's view, or may enhance face-to-face conversation by producing shared interactive virtual models (Rekimoto 1996. In this way Mixed Reality techniques can be used to enhance communication regardless of proximity. Thus the use of Mixed Reality facilitates the development of collaborative interfaces that go "beyond being there". By using see-through head mounted displays with head and body tracking in a collaborative interface. These types of Mixed Reality interfaces allow multiple users in the same location or remote to work in both the real and virtual world simultaneously, facilitating CSCW in a seamless manner.

According to (Schmalsteig et al. 1996), there are five key advantages of collaborative MR environments:

- Virtuality: Objects that don’t exist in the real world can be viewed and examined.
- Augmentation: Real objects can be augmented by virtual annotations.
- Cooperation: Multiple users can see each other and cooperate in a natural way.
- Independence: Each user controls his own independent viewpoint.
- Individuality: Displayed data can be different for each viewer.

1.6 Conclusion

In this chapter, we have defined the key concepts related to virtual reality and augmented reality. We have found that over time several definitions have been proposed for these two terms.

Also, we have presented the various tasks and techniques of 3D interaction. However, we have found that most of the existing interaction techniques rely on the use of hand gestures as interaction tools. Therefore, in the next chapter we will discuss the different techniques for recognizing existing gestures.

Chapter 2. Gesture Recognition

Chapter 2 Gesture Recognition

2.1 Introduction

Since the advent of computing, technology has moved toward a more natural user interface. From the invention of early graphic user interfaces (GUIs), and mice to interact with them, humans have worked toward more instinctive ways to interface with the machines they operate. With many advances in technical ingenuity, ideas and creative conceptions can outpace the technology needed to bring those ideas to fruition.

While we do not yet have seamless gesture control integrated into our daily lives, we are closer than many realize to living in a world where humans can interact with computers as they would interact with each other. Adding support for various types of gestures toward electronic devices enables us to use our natural "language" to operate these devices, which is much more intuitive and effortless when compared to touching a screen, manipulating a mouse or remote control, tweaking a knob or pressing a switch.

2.2 Definitions

2.2.1 Gesture

The word gesture can refer to any non-verbal communication that is intended to communicate a specific message. In the world of gesture recognition, a gesture is defined as any physical movement, large or small, that can be interpreted by a motion sensor. It may include anything from the

pointing of a finger to a roundhouse kick or a nod of the head to a pinch or wave of the hand.

2.2.2 Gesture recognition

According to (marxentlabs 2020), Gesture recognition is a type of perceptual computing user interface that allows computers to capture and interpret human gestures as commands. The general definition of gesture recognition is the ability of a computer to understand gestures and execute commands based on those gestures.

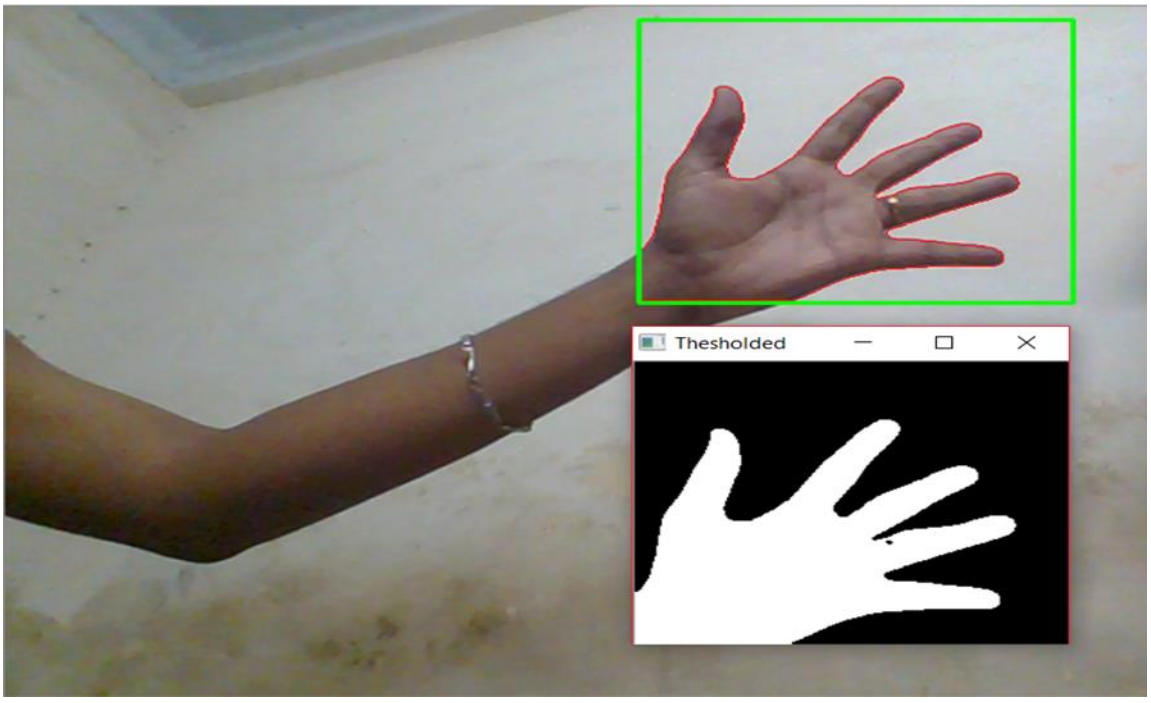


Figure 2-1 Gesture recognition illustration.

Gesture recognition is an alternative user interface for providing real-time data to a computer. Instead of typing with keys or tapping on a touch screen, a motion sensor perceives and interprets movements as the primary source of data input. This is what happens between the time a gesture is made and the computer reacts.

- A camera feeds image data into a sensing device that is connected to a computer. The sensing device typically uses an infrared sensor or projector for the purpose of calculating depth,
- Specially designed software identifies meaningful gestures from a predetermined gesture library where each gesture is matched to a computer command.
- The software then correlates each registered real-time gesture, interprets the gesture and uses the library to identify meaningful gestures that match the library.
- Once the gesture has been interpreted, the computer executes the command correlated to that specific gesture.

2.3 Gesture Recognition applications

2.3.1 Medicine and Assistive Technologies

Hand gestures can be utilized to interact with visualization screens, control medical devices, and assist handicapped persons as part of their rehabilitation treatment. Some of these ideas have been used to enhance medical procedures and systems. In (Shamayleh et al. 2018), methods were discussed to integrate

hand gestures with physician-computer interfaces, illustrating a computer-vision system that allows doctors to carry out standard mouse functions such as pointer movement and button presses with hand gestures that meet with the "intuitiveness" requirement. In (Cote et al. 2006), a hand-gesture-tracking machine named Gestix was designed, which permits doctors to explore MRI images in an operating room via a natural interface to meet with both "come as you are" and "intuitiveness".



Figure 2-2 Surgeon using Gestix to browse medical images (Cote et al. 2006).

2.3.2 Gaming (Entertainment)

Computer games are a technologically promising and commercially rewarding field for innovative interfaces because of the entertaining nature of the interaction. People are willing to try new interface technologies because they have the chance to be immersed in a challenging game-like environment (Zabulis et al. 2009). In computer-vision-based hand gesture- controlled games, the system has to react fast to user gestures, must be robust and efficient in contrast to other applications.

The apparition of reasonably inexpensive image and depth sensors has encouraged researchers in the area of object detection, tracking and gesture recognition. A popular device is the Kinect, developed by Microsoft for the Xbox 360 video game console. It employs cameras and microphones to detect and track the motions of a user standing in front of it and permits voice control of games and video content. Using a webcam-style add-on peripheral for the Xbox 360 console, it permits players to control 31 and communicate with the Xbox 360 by a natural user interface using gestures and voice commands without the requirement to touch a game controller. The Kinect is capable of concurrently tracking up to six players, involving two active users for movement analysis with a feature extraction of 20 joints per user. Although the Kinect can detect hand gestures, it does not come with a hand gesture recognition application software. Its software permits advanced human body tracking, facial recognition and voice recognition.



Figure 2-3 Playing bowling using The Kinect device.

2.3.3 Human Robot Interaction

Another traditional topic of application for gesture interaction systems is human-robot interaction with a huge number of published papers. Hand gesture recognition is a significant feature for fixed and portable robots, robotic manipulators and human users were integrated with hand gesture commands for recognizing four postures such as when a user points at an object, the robot can detect it., a human-machine interaction system for game playing allows the visual recognition of three postures with different rotation angles and scales, a robotic- assistant interaction application was designed by combining gesture recognition and voice to generate commands. Once the hand posture is detected, six gestures are trained using a hand contour as the major feature of each gesture and final, a human-robot system was designed to recognize dynamic gestures by tracking head and arm direction.

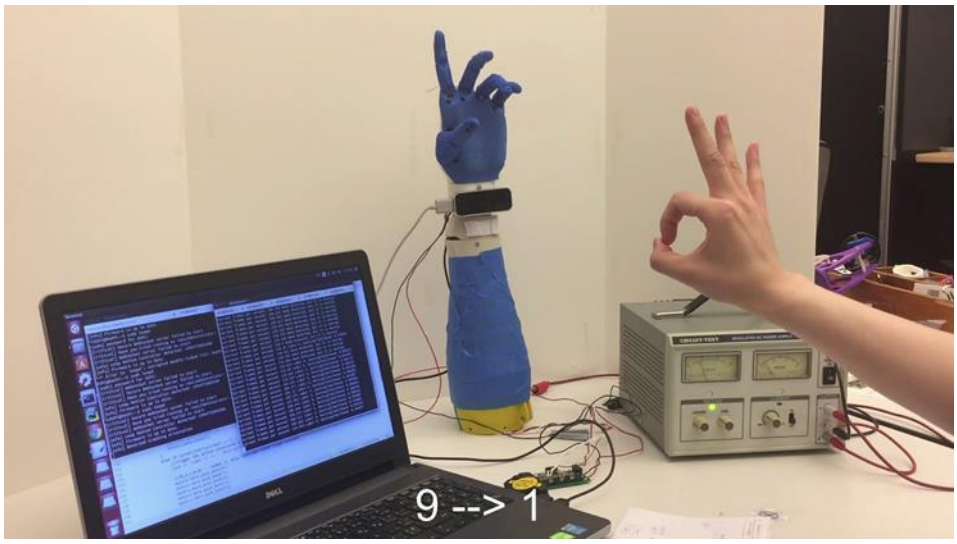


Figure 2-4 Interaction with a robotic arm.

2.4 Vision based gesture recognition

The techniques for gesture recognition vary from geometric template matching, warping the time dimension, boundary tests for classification, statistical models, and many others that are not mentioned here. These techniques suffer from different problems of the gesture recognition domain.

(Shamayleh et al. 2018) Most of the complete hand interactive mechanisms that act as a building block for vision based hand gesture recognition system are comprised of three fundamental phases: detection, tracking and recognition as shown in the figure below (figure 2-5).

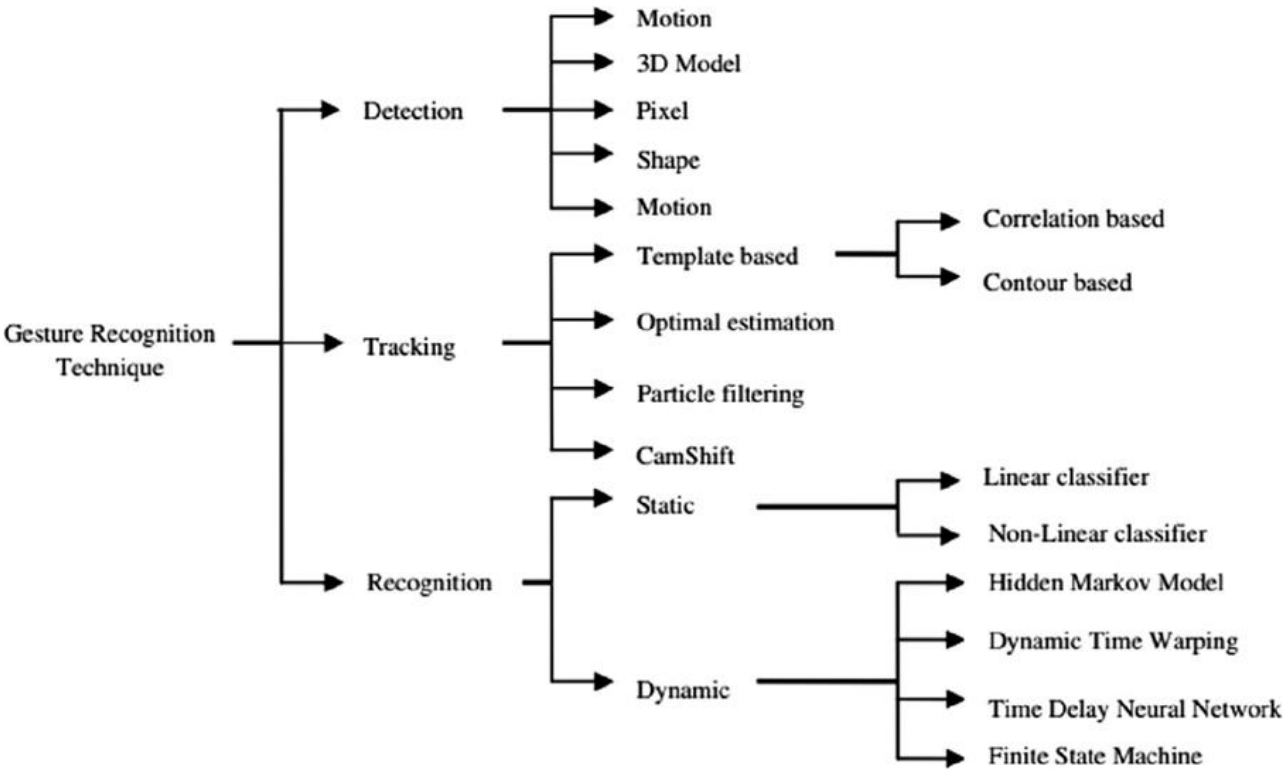


Figure 2-5 Vision based hand gesture recognition techniques (Shamayleh et al. 2018).

2.4.1 Detection

The primary step in hand gesture recognition systems is the detection of hands and the segmentation of the corresponding image regions. This segmentation is crucial because it isolates the task-relevant data from the image background, before passing them to the subsequent tracking and recognition stages. A large number of methods have been proposed in the literature that utilize a several types of visual features and, in many cases, their combination. Such features are skin color, shape, motion and anatomical models of hands. In (Cote et al. 2009) and (Zabulis et al. 2009) a comparative study on the performance of some hand segmentation techniques can be found.

2.4.1.1 Color

Skin color segmentation has been utilized by several approaches for hand detection. A major decision towards providing a model of skin color is the selection of the color space to be employed. Several color spaces have been proposed including RGB, normalized RGB, HSV, YCrCb, YUV, etc. Color spaces efficiently separating the chromaticity from the luminance components of color are typically considered preferable. This is due to the fact that by employing chromaticity-dependent components of color only, some degree of robustness to illumination changes can be achieved. (Terrillon et al. 2000) review different skin chromaticity models and evaluate their performance.

To increase invariance against illumination variability some methods operate in the HSV, YCrCb or YUV colorspaces, in order to approximate the "chromaticity" of skin rather than its apparent color value. They typically eliminate the luminance component, to remove the effect of shadows, illumination changes, as well as modulations of orientation of the skin surface relative to the light source(s). The remaining 2D color vector is nearly constant for skin regions and a 2D histogram of the pixels from a region containing skin shows a strong peak at the skin color. The perceived color of human skin varies greatly across human races or even between individuals of the same race. Additional variability may be introduced due to changing illumination conditions and/or camera characteristics. Therefore, color-based approaches to hand detection need to employ some means for compensating for this variability. In (Sigal et al. 2004) an invariant representation of skin color against changes in illumination is pursued, but still with not conclusive results. Another adaptation technique estimates the new parameters for the mean and

covariance of the multivariate Gaussian skin color distribution, based on a linear combination of previous parameters. However, most of these methods are still sensitive to quickly changing or mixed lighting conditions. In general, color segmentation can be confused by background objects that have a color distribution similar to human skin. A way to cope with this problem is based on background subtraction (Rehg & Kanade 1994). However, background subtraction is typically based on the assumption that the camera system does not move with respect to a static background. To solve this problem, some research looked into the dynamic correction of background models and/or background compensation methods.

2.4.1.2 Shape

The characteristic shape of hands has been utilized to detect them in images in multiple ways. Much information can be obtained by just extracting the contours of objects in the image. If correctly detected, the contour represents the shape of the hand and is therefore not directly dependent on viewpoint, skin color and illumination. On the other hand, the expressive power of 2D shape can be hindered by occlusions or degenerate viewpoints. In the general case, contour extraction that is based on edge detection results in a large number of edges that belong to the hands but also to irrelevant background objects. Therefore, sophisticated post-processing approaches are required to increase the reliability of such an approach. In this spirit, edges are often combined with skin color and background subtraction motion cues.

2.4.1.3 Pixel values

Significant work has been carried out on finding hands in grey level images based on their appearance and texture. In (Wu & Huang 2000), the suitability of a number of classification methods for the purpose of view-independent hand posture recognition was investigated. Several methods attempt to detect hands based on hand appearances, by training classifiers on a set of image samples. The basic assumption is that hand appearance differs more among hand gestures than it differs among different people performing the same gesture. Still, automatic feature selection constitutes a major difficulty. Moreover, methods based on a machine learning approach called boosting have demonstrated very robust results in face and hand detection. Due to these results, they are reviewed in more detail below. Boosting is a general method that can be used for improving the accuracy of a given learning algorithm (Schapire 2002). It is based on the principle that a highly accurate or "strong" classifier can be derived through the linear combination of many relatively inaccurate or\weak" classifiers. In general, an individual weak classifier is required to perform only slightly better than random. As proposed in (Viola & Jone 2001) for the problem of hand detection, a weak classifier might be a simple detector based on basic image block differences efficiently calculated using an integral image.

2.4.1.4 3D Model

A category of approaches utilize 3D hand models for the detection of hands in images. One of the advantages of these methods is that they can achieve view-independent detection.

The employed 3D models should have enough degrees of freedom to adapt to the dimensions of the hand(s) present in an image. Different models require different image features to construct feature-model correspondences. Point and line features are employed in kinematic hand models to recover angles formed at the joints of the hand.

Hand postures are then estimated provided that the correspondences between the 3D model and the observed image features are well established. Various 3D hand models have been proposed in the literature. Some approaches utilize a deformable model framework to fit a 3D model of the hand to image data. The fitting is guided by forces that attract the model to the image edges, balanced by other forces that tend to preserve continuity and evenness among surface points. In (Lee & Kunii 1995), the process is enhanced with anatomical data of the human hand that are incorporated into the model. Also, to fit the hand model to an image of a real hand, characteristic points on the hand are identified in the images, and virtual springs are implied which pull these characteristic points to goal positions on the hand model.

2.4.1.5 Motion

Motion is a cue utilized by a few approaches to hand detection. The reason is that motion based hand detection demands for a very controlled setup, since

it assumes that the only motion in the image is due to hand movement. Indeed, early works (Cui & Weng 1996) assumed that hand motion is the only motion occurring in the imaged environment. In more recent approaches, motion information is combined with additional visual cues. In the case of static cameras, the problem of motion estimation reduces to that of background maintenance and subsequent subtraction. For example in (Martin et al. 1998) such information is utilized to distinguish hands from other skin-colored objects and cope with lighting conditions imposed by colored lights. The difference in luminance of pixels from two successive images is close to zero for pixels of the background. By choosing and maintaining an appropriate threshold, moving objects are detected within a static scene, in (Yuan et al. 1995) a novel feature, based on motion residue, is proposed. Hands typically undergo non-rigid motion, because they are articulated objects. Consequently, hand detection capitalizes on the observation that for hands, interframe appearance changes are more frequent than for other objects such as clothes, face, and background.

2.4.2 Tracking

If the detection method is fast enough to operate at image acquisition frame rate, it can be used for tracking as well. However, tracking hands is notoriously difficult since they can move very fast and their appearance can change vastly within a few frames. Tracking can be defined as the frame-to-frame correspondence of the segmented hand regions or features towards understanding the observed hand movements. The importance of robust tracking is twofold. First, it provides the inter-frame linking of hand/finger

appearances, giving rise to trajectories of features in time. These trajectories convey essential information regarding the gesture and might be used either in a raw form (e.g. in certain control applications like virtual drawing the tracked hand trajectory directly guides the drawing operation) or after further analysis (e.g. recognition of a certain type of hand gesture). Second, in model-based methods, tracking also provides a way to maintain estimates of model parameters variables and features that are not directly observable at a certain moment in time.

2.4.2.1 Template Based

This class of methods exhibits great similarity to methods for hand detection. Members of this class invoke the hand detector at the spatial vicinity that the hand was detected in the previous frame, so as to drastically restrict the image search space. The implicit assumption for this method to succeed is that images are acquired frequently enough.

2.4.2.1.1 Correlation-based feature tracking

In (Crowley et al. 1995), correlation-based template matching is utilized to track hand features across frames. Once the hand(s) have been detected in a frame, the image regions in which they appear is utilized as the prototype to detect the hand in the next frame. Again, the assumption is that hands will appear in the same spatial neighborhood. This technique is employed for a static camera in (Darrell et al. 1996), to obtain characteristic patterns or signatures of gestures, as seen from a particular view. The work in (Cootes &

Taylor 1992) deals also with variable illumination. A target is viewed under various lighting conditions. Then, a set of basis images that can be used to approximate the appearance of the object viewed under various illumination conditions is constructed. Tracking simultaneously solves for the affine motion of the object and the illumination. Real-time performance is achieved by pre-computing "motion templates" which are the product of the spatial derivatives of the reference image to be tracked and a set of motion fields. Some approaches detect hands as image blobs in each frame and temporally correspond blobs that occur in proximate locations across frames. Approaches that utilize this type of blob tracking are mainly the ones that detect hands based on skin color, the blob being the correspondingly segmented image region. Blob-based approaches are able to retain tracking of hands even when there are great variations from frame to frame.

2.4.2.1.2 Contour based tracking:

Deformable contours or "snakes" have been utilized to track hand regions in successive image frames (Cootes & Taylor 1992). Typically, the boundary of this region is determined by intensity or color gradient. Nevertheless, other types of image features (e.g. texture) can be considered. The technique is initialized by placing a contour near the region of interest. The contour is then iteratively deformed towards nearby edges to better fit the actual hand region. This deformation is performed through the optimization of energy functional that sums up the gradient at the locations of the snake while, at the same time, favoring the smoothness of the contour. When snakes are used for tracking, an active shape model is applied to each frame and the convergence of the snake

in that frame is used as a starting point for the next frame. Snakes allow for real-time tracking and can handle multiple targets as well as complex hand postures. They exhibit better performance when there is sufficient contrast between the background and the object. On the contrary, their performance is compromised in cluttered backgrounds. The reason is that the snake algorithm is sensitive to local optima of the energy function, often due to ill foreground/background separation or large object displacements and/or shape deformations between successive images.

2.4.2.2 Optimal estimation

Feature tracking has been extensively studied in computer vision. In this context, an optimal estimation framework had been widely employed in turning observations (feature detection) into estimations (extracted trajectory). The reasons for its popularity are real-time performance, treatment of uncertainty, and the provision of predictions for the successive frames. In (Argyros & Lourakis 2004), the target is retained against cases where hands occlude each other, or appear as a single blob in the image, based on a hypothesis formulation and validation/rejection scheme. The problem of multiple blob tracking was investigated in (Argyros & Lourakis 2004), where blob tracking is performed in both images of a stereo pair and blobs are corresponded, not only across frames, but also across cameras. The obtained stereo information not only provides the 3D locations of the hands, but also facilitates the potential motion of the observing stereo pair which could be thus mounted on a robot that follows the user.

2.4.2.3 Particle filtering

Particle filters have been utilized to track the position of hands and the configuration of fingers in dense visual clutter. In this approach, the belief of the system regarding the location of a hand is modeled with a set of particles. The approach exhibits advantages over Kalman filtering, because it is not limited by the unimodal nature of Gaussian densities that cannot represent simultaneous alternative hypotheses. A disadvantage of particle filters is that for complex models such as the human hand many particles are required, a fact which makes the problem intractable especially for high-dimensional models. Therefore, other assumptions are often utilized to reduce the number of particles.

The CONDENSATION algorithm (Isard & Blake 1998) which has been used to learn to track curves against cluttered backgrounds, exhibits better performance than Kalman filters, and operates in real-time. It uses "factored sampling", previously applied to the interpretation of static images, in which the probability distribution of possible interpretations is represented by a randomly generated set. Condensation uses learned dynamical models, together with visual observations, to propagate this random set over time. The result is highly robust tracking of agile motion, extending the CONDENSATION algorithm the work in (Mammen et al. 2001) detects occlusions with some uncertainty. Another work used the same algorithm is integrated with color information; the approach is based on the principle of color histogram distance, but within a probabilistic framework, the work in introduces a new Monte Carlo tracking technique. In general, contour tracking techniques, typically, allow

only a small subset of possible movements to maintain continuous deformation of contours.

2.4.2.4 CamShift

This algorithm is based on the principle of mean shift algorithm. Mean-shift is a kernel based tracking method which uses density-based appearance models to represent targets. The method tracks targets by finding the most similar distribution pattern in a frame sequences with its sample pattern by iterative searching. It has been widely used because of its relative simplicity and low computational cost, but mean-shift would fail in changing the track window‘s scale, as targets move toward or away from the camera (Yilmaz et al. 2006) Based on mean-shift, continuous adaptive mean-shift (CamShift) was proposed to overcome the problem. CamShift adaptively adjusts the track window‘s size and the distribution pattern of targets during tracking. CamShift algorithm can be used to track the distribution of any kind of feature that represents the target in a lightweight, robust and efficient way (Bradski & Kaehler 2008). Most researchers though, use color data to represent targets for CamShift, this would give a low complexity and practical performance to the method. While CamShift performs well with objects that have a simple and constant appearance, it is not robust in more complex scenes.

2.4.3 Recognition

The overall goal of hand gesture recognition is the interpretation of the semantics that the hand(s) location, posture, or gesture conveys.

Several representations have been proposed in order to recognize and model a gesture from the hand. According to (Rautaray & Agrawal 2012), two main categories of representation of hand gestures exist: methods based on 3D models and methods based on appearance, as shown in the figure below (figure 2-6)

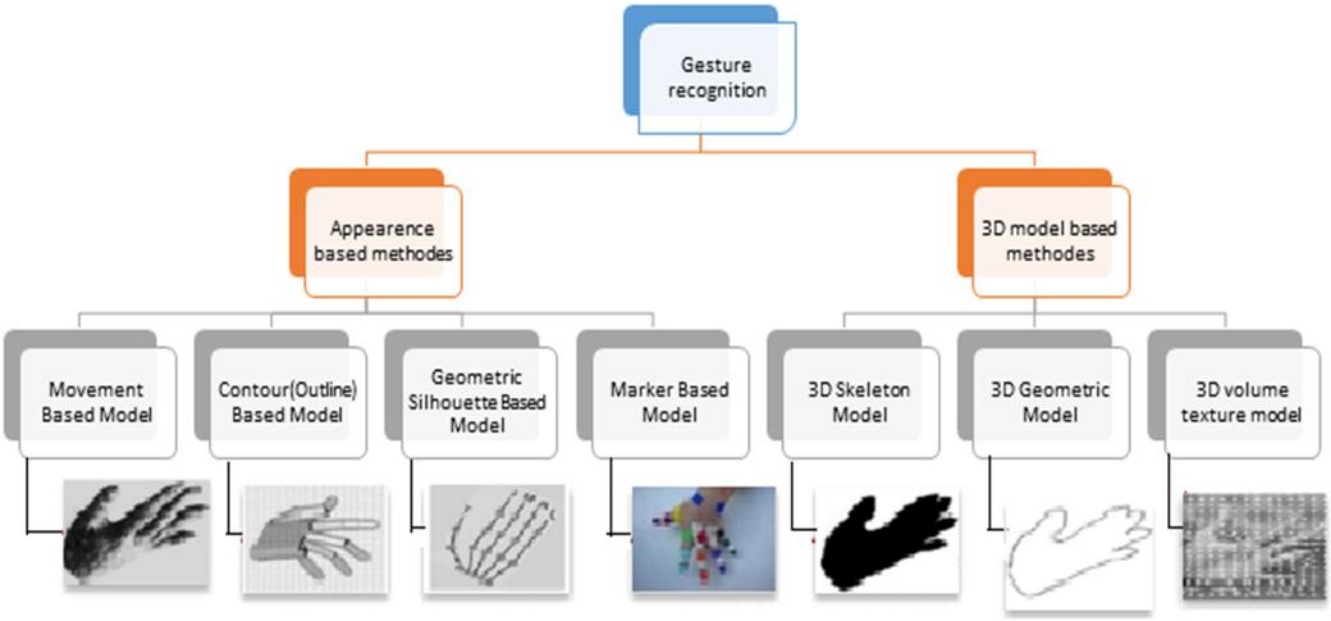


Figure 2-6 Hand Gesture representation methods (Rautaray & Agrawal 2012).

2.4.3.1 Appearance based techniques

Some approaches have been proposed, the principle of which is to wear easily recognizable markers on the hands (gloves, colors, coded targets, etc.) in order to interact with the system (Benbelkacem et al. 2012). In (Wang & Popović 2009), a 3D hand gesture recognition technique while wearing colored gloves was proposed. The colored gloves are modeled and constructed to

facilitate the recognition of dynamic hand gestures by a camera (RGB) to provide the user with a natural interaction with the virtual environment. Although the results shown by the authors are convincing, however, this system is designed to recognize this form of gloves.



Figure 2-7 Colored gloves technique by (Wang & Popović 2009).

(Bellarbi et al. 2011) Proposed to wear small colored markers on the fingers of the user's hands in order to manipulate digital documents projected on a table. The markers are made with pieces of colored paper so as not to clutter the user. However, this technique makes it possible to recognize a limited number of static gestures for an interaction in 2D, and presents certain anomalies when the colors worn are confused with those of the scene.

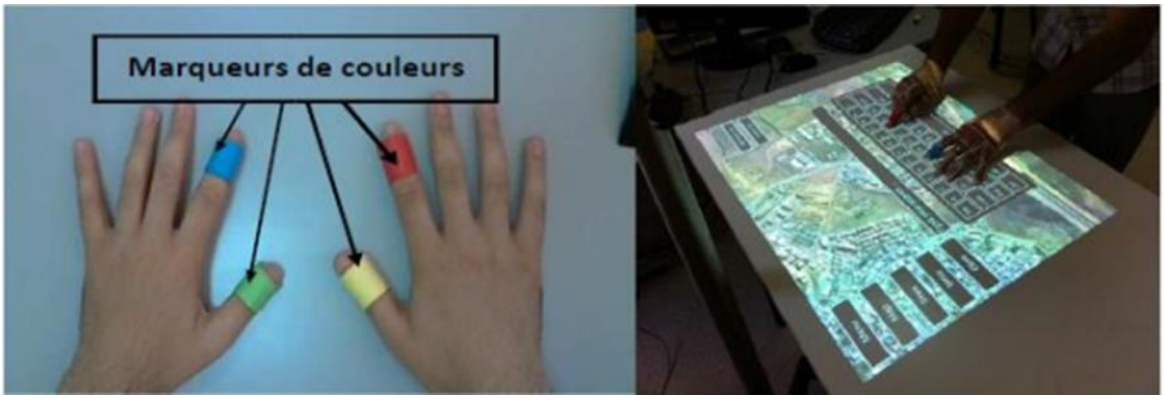


Figure 2-8 Interaction technique based on color marker by (Bellarbi et al. 2011).

On the other hand, many approaches have been proposed in order to interact naturally with the machine using the hand without having to wear a marker or a sensor on the user's hand and this by processing images captured by the camera. Some approaches have considered the problem of gesture recognition as a classification problem using classification techniques.

In this sense, neural networks have been widely used for the recognition of gestures (Stergiopoulou & Papamarkos 2009) and (Hasan & Abdul-Kareem 2014). For their part, the hidden Markov chains (HMM) were also exploited for the classification of gestures. We can cite (Jinjun et al. 2009) and (Fahn & Chu 2011). However, these approaches do not guarantee real-time applications. Likewise, certain approaches have been proposed for the recognition of gestures, using invariant moments (Liu et al. 2012), Krawtchouk moments (Priyal & Bora 2013), Support Vector Machine SVM,Gabor Filter (Huang et al. 2011) Dynamic Time Warpping DTW, Template Matching or even Randomized Decision Forests.

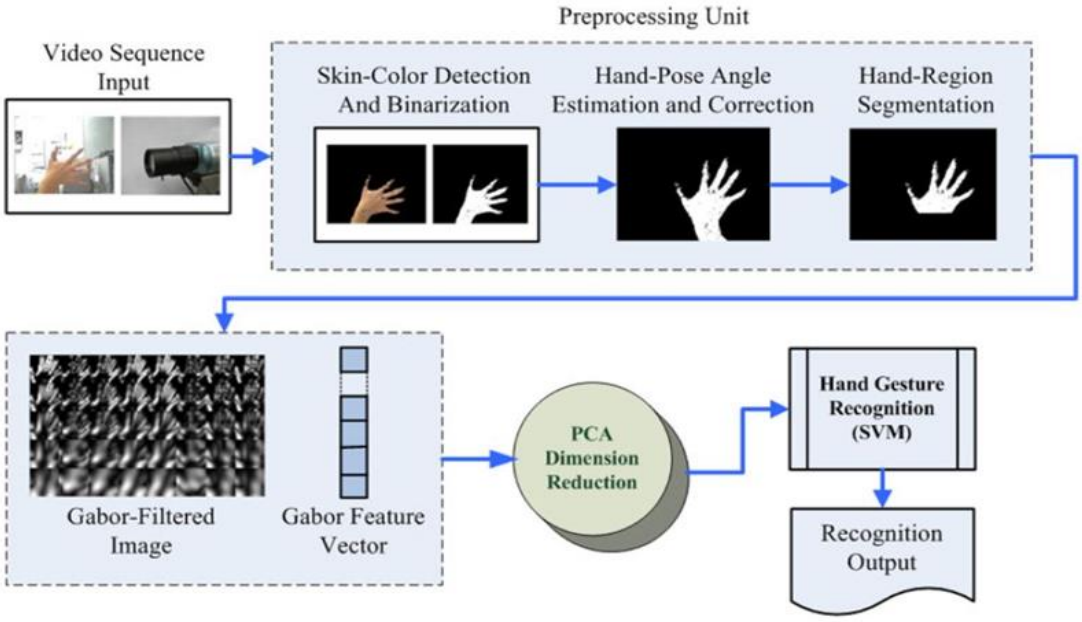


Figure 2-9 Flow diagram of the gesture recognition technique by Gabor filter proposed in (Huang et al. 2011).

Other approaches have applied point-of-interest descriptors to captured images, to recognize hand gestures, such as SIFT (Lin et al. 2013), SURF (Yao & Li 2013), LBP (Ding et al. 2011). (Bellarbi et al. 2013) proposed a technique for the detection and recognition of 2D hand gestures using a simple webcam. The proposed technique is based on the one hand on adaptive binarization and the computation of the histogram of the image for the detection and extraction of the hand, and on the other hand on the Chain Code algorithm with a modified version of the ASM (Approximate String Matching) technique for gesture recognition. This technique was used to interact in 2D with an interactive table.



Figure 2-10 The interactive table proposed by (Bellarbi et al. 2013).

2.4.3.2 3D Model Based Techniques

By exploiting the advantages of depth cameras, such as Microsoft's Kinect (Wikipedia 2020c), and Asus's Xtion (asus 2020) (see figure 2-11), several approaches to 3D gesture recognition have been proposed.



Figure 2-11 Some of Depth camera technologies.

Wang and associates (Wang et al. 2011) proposed a technique called 3Gear. The idea for this technique is based on searching a database of gestures to find the gesture most similar to the gesture captured by the Kinect (figure below).

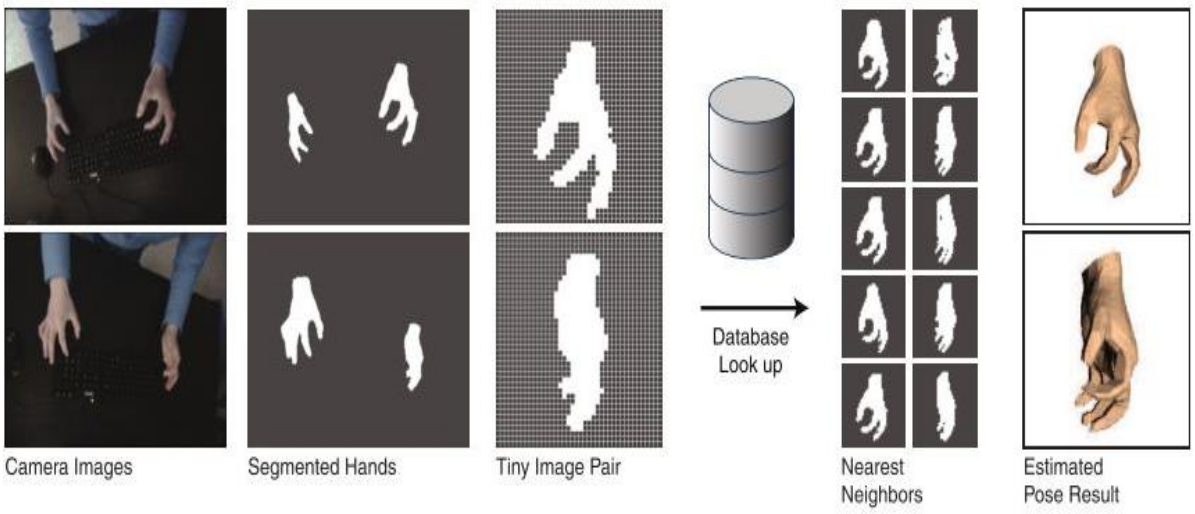


Figure 2-12 The Principle of 3Gear technique (Wang et al. 2011).

The gesture database generated offline contains images resulting from the projection of different possible gestures from a 3D model of the hand. This technique features robust recognition of 3D hand gestures.

Messaci and associates (Messaci et al. 2015) used this technique to interact in a virtual environment. The objective is to train technicians in the field of maintenance (figure below).



Figure 2-13 Example of a user who operates in a virtual environment, using the system proposed in (Messaci et al. 2015).

With an almost similar idea, Khamis and associates (Khamis et al. 2015) proposed a machine learning algorithm for efficient search of a database of depth images of gestures. This technique has shown better results. However, it is more expensive in terms of computation time.

Some researchers have modeled the hand according to points of articulation. They then applied a cost function, to minimize the error between the 3D model and the shape of the real hand, in order to estimate the position and 3D gesture of the hand (Figure 2-14). We can cite the works (Taylor et al. 2016), (Qian et al. 2014) and (Melax et al. 2013).

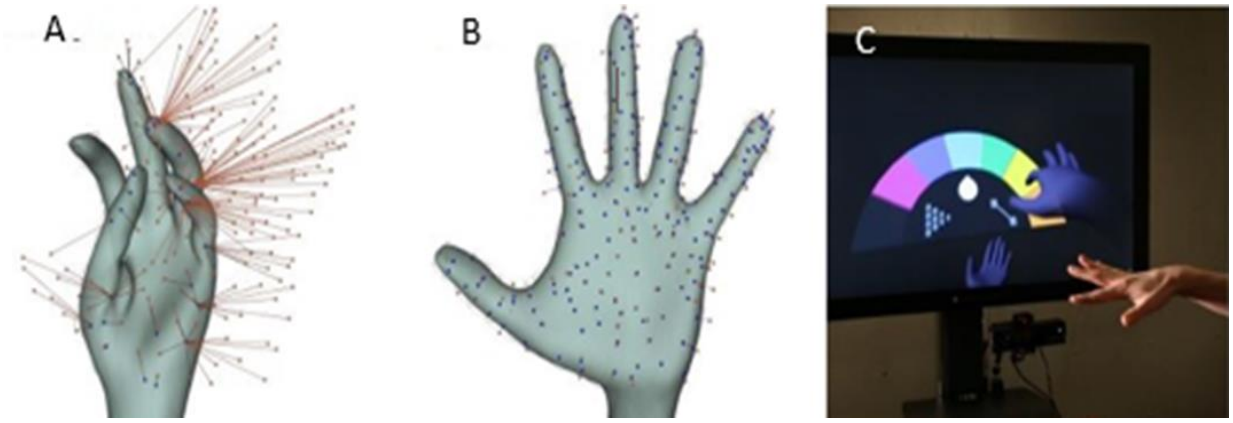


Figure 2-14 Hand gesture recognition and tracking technique proposed in (Taylor et al. 2016). a) Initialization: estimated model of gesture. b) Convergence towards the real gesture by applying an energy optimization function. c) Example of VR application.

Recently, 3D gesture recognition techniques have taken on another dimension in terms of robustness and precision with the evolution of learning techniques and the emergence of Deep Learning. We can cite in this sense the works (Wu et al. 2016), (Tompson et al. 2014) and (Oberweger et al. 2015).

These techniques show efficiency and robustness for the recognition of 3D hand gestures. However, the capture devices in use today require a fixed position of the user, and do not offer full mobility. Recent comparative studies on the different techniques of recognition of 3D hand gestures can be found in (Cheng et al. 2016) and (Supancic et al. 2015).

In October 2012, Holz and associates (Holz et al. 2012) developed the "Leap Motion", a new device for recognizing and tracking hand gestures, consisting of a soft part and a controller (hardware). The latter, which is 13 mm x 30 mm x 76 mm in size, is equipped with two infrared cameras (figure 2-15) and can detect and recognize the hands in a space that can go up to 50 cm.



Figure 2-15 Leap motion.

2.5 Conclusion

In this chapter, we dove into the gesture recognition, we mentioned a few of its applications and technologies in fields such as Medicine and Human-Robot interaction. In addition, we gave a detailed view for the vision based gesture recognition which includes Detection, Tracking and Recognition.

To conclude, we have seen that 3D interaction is an interesting tool for human-machine interaction in general and for interfaces based on Mixed Reality in particular, because these technological advances allow the user to interact with a system naturally or with specific devices. Furthermore, with the development of gesture recognition technology, this interaction would be available with minimal equipment.

Although a lot of work has been done in this field, it is still a topic of current research and researchers are still hoping for better results in the future.



Part two: Conception and implementation



Chapter 3. Conception and analysis

Chapter 3 Conception and analysis

3.1 Introduction

Analysis and conception are processes that aim to formalize the preliminary stages in the development of a system in order to best respond to the various problems posed.

In This chapter we will present the analysis we performed through a description of the system, the technical and functional specifications of the latter. Next, we will detail the design using functional diagrams that we have designed.

3.2 Description of the system

As we presented in the general introduction, we aim to develop an interactive and collaborative augmented reality system based on gesture recognition using neural networks. This system should allow users to share and visualize virtual objects by augmented reality, as well as interact in 3D and co-manipulate shared objects using gesture recognition.

For this, this system is made up of at least two terminals (or more), in our case, we have set up a Windows laptop and an Android smartphone. The laptop has two cameras, the front camera used for gesture recognition, the rear one for augmented reality.

The following schema shows an overall description of this system:



Figure 3-1 Overall presentation of the proposed system

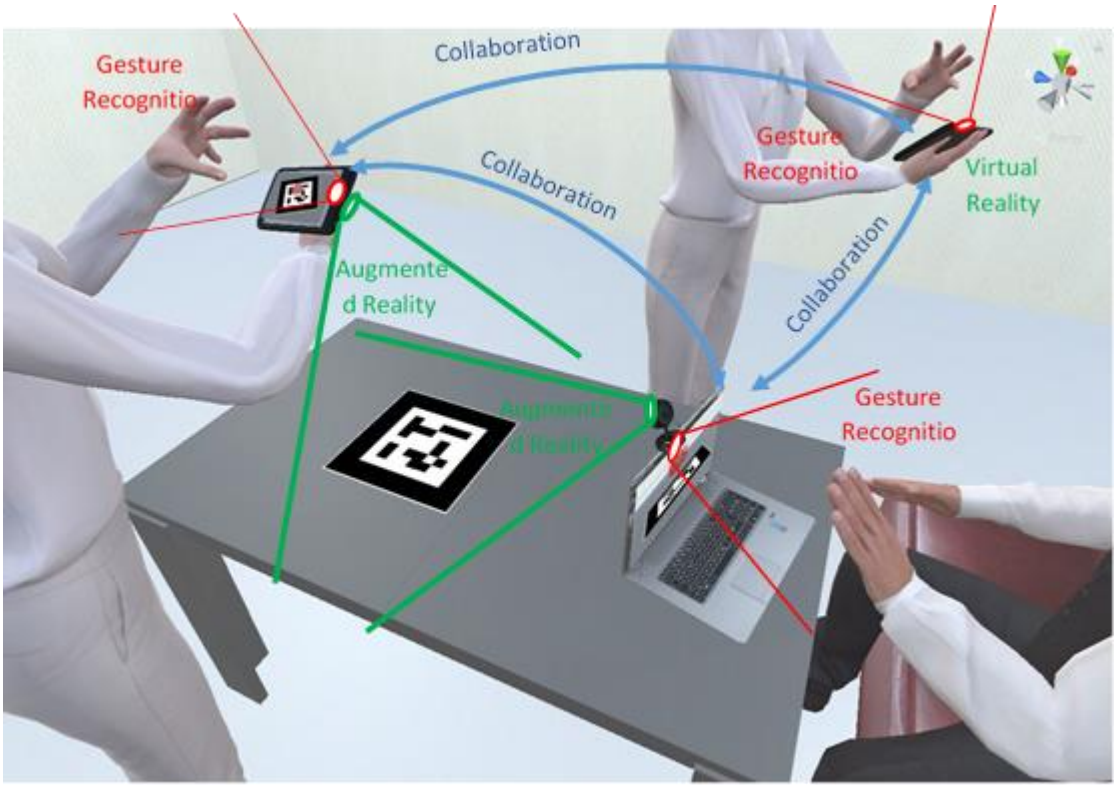


Figure 3-2 Descriptive schema of the proposed system.

3.3 System specification

In order to identify all the functionalities that our application can offer and the constraints that it must respect, we list in the following its functional and technical specifications.

3.3.1 Functional Specifications

1) The system must display an AR or VR interface on the user's screen.
2) The system must recognize some predefined gestures made by the user (minimum two gestures).
3) The system must follow the movements of the user's hand.
4) The system must allow the user to select virtual objects.
5) The system must allow the user to manipulate virtual objects (translation + rotation) using hand gestures.
6) The system must allow users to connect with each other users and share their 3D scenes.
7) The system must allow each user to see the manipulations made by other users in real time.

3.3.2 Technical specifications

1) The system must be multiplatform (Android, Windows).
2) The system must allow connection between different platforms.
3) The Android app must use the front camera for gesture recognition and the back camera for AR (if the smartphone allows it).

4) Windows application must use the front camera for gesture recognition and the rear camera for AR.

5) The system must be developed in C # under Unity 3D together with Microsoft Visual Studio.

3.4 System architecture

We can describe our system in the form of three (3) main Modules, namely: Gesture recognition module by neural networks, Mixed Reality (RM) module (VR / AR) and the Collaboration and data sharing module.

3.4.1 Neural network gesture recognition module

As shown in the figure (figure 3-2), this module provides gesture recognition by the following steps:

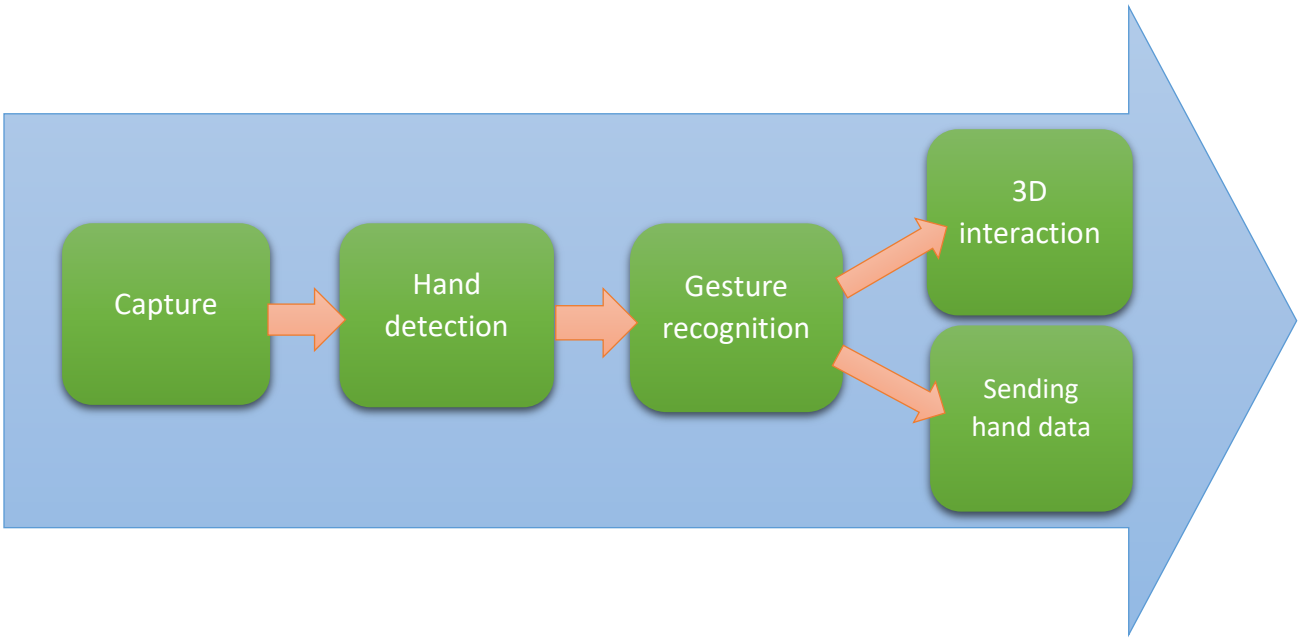


Figure 3-3 Brave descriptive schema of the gesture recognition module.

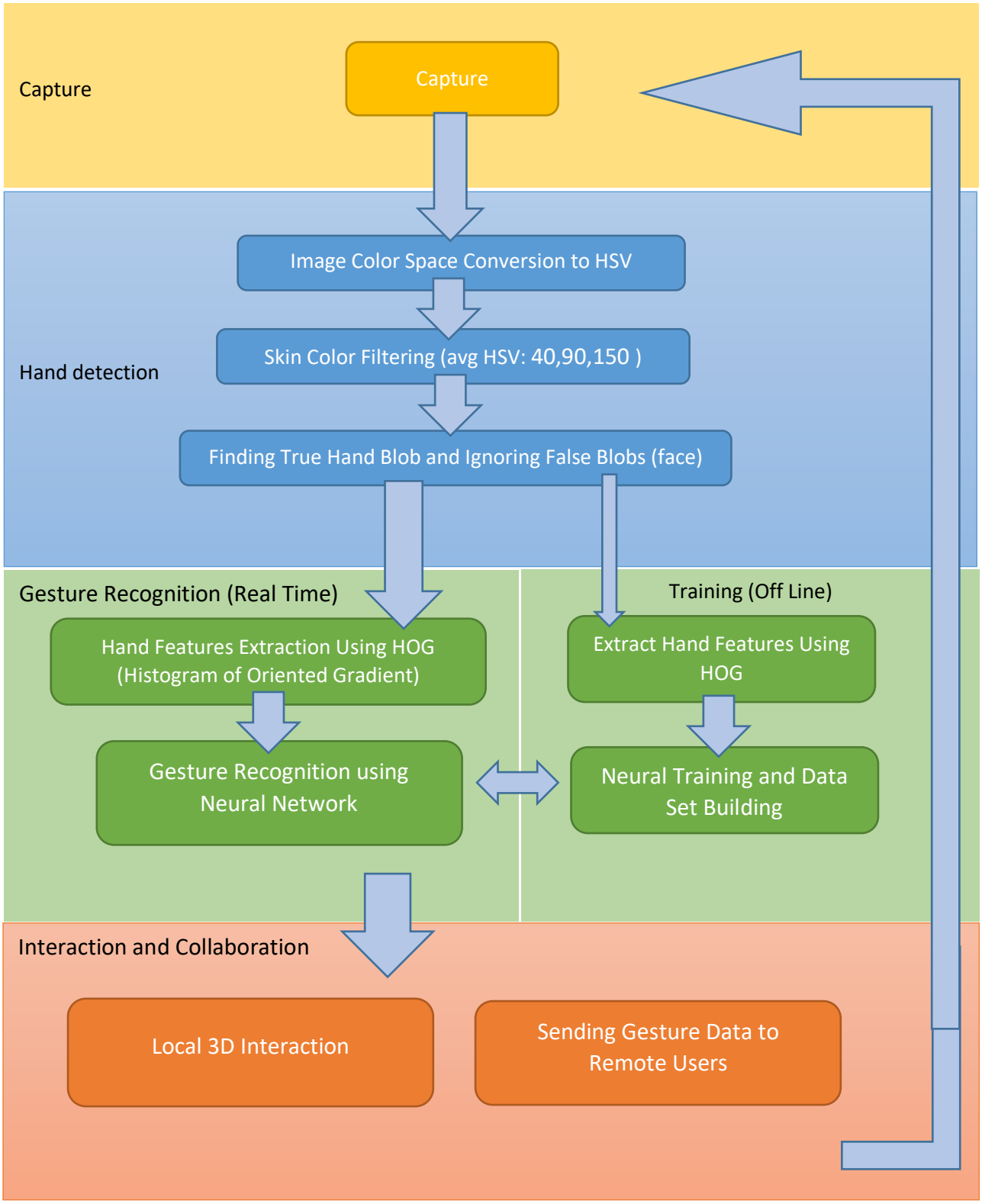


Figure 3-4 Descriptive schema neural network gesture recognition module.

This recognition process goes through these steps:

A. Capture:

The system captures images using the front smartphone camera or laptop.

B. Hand Detection:

In order to detect the user hands, we first converted the image color space from RGB to HSV (Hue Saturation Value), this color (advantages de HSV par rapport RGB) the big advantage of this color space is this it is less sensitive to shadow and uneven lighting (BELLARBI et al.2011) . A graphical representation is given in figure 3-4.

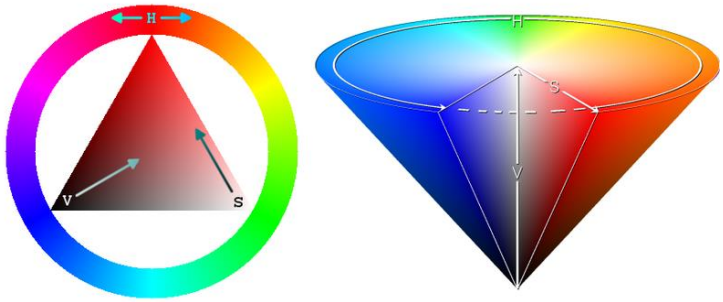


Figure 3-5 HSV color space.

From the converted HSV image, we applied a Skin Color filtering, the standard skin color range is (KOLKUR et al.2017):

Min: (HSV: 0, 30, 60).

Max: (HSV: 80,150,255).

However, this color range may filter other objects than Hands, such as faces or other objects within the same color range.

In the other hand, this filtering step doesn't extract solid hand blob, so the detected blobs are separated. Hence, for tackling those problems, we applied an algorithm for max area detection using assembled contours and convex hull.

```
1 reference
public void ColorFilterHsv(Scalar colorMin, Scalar colorMax) {

    Imgproc.cvtColor(imageRgb, imageHsv, Imgproc.COLOR_RGB2HSV_FULL);
    Core.inRange(imageHsv, colorMin, colorMax, imageBlobHsv);
    Imgproc.dilate(imageBlobHsv, imageBlobHsv, new Mat());
    Imgproc.GaussianBlur(imageBlobHsv, imageBlobHsv, new Size(5, 5), 3);
    Imgproc.GaussianBlur(imageBlobHsv, imageBlobHsv, new Size(5, 5), 3);
    Imgproc.GaussianBlur(imageBlobHsv, imageBlobHsv, new Size(5, 5), 3);
    List<MatOfPoint> contours = new List<MatOfPoint>();

    Imgproc.findContours(imageBlobHsv, contours, mHierarchy, Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE);
    Scalar color = new Scalar(0, 255, 0);


    double maxArea = 0;
    int contourIndex = 0;
    for (int i = 0; i < contours.Count; i++)
    {

        MatOfPoint wrapper = contours[i];
        double area = Imgproc.contourArea(wrapper);

        if (area > maxArea)
        {

            contourIndex = i;
            maxArea = area;

        }

    }
```

Figure 3-6 Method used for hand detection.

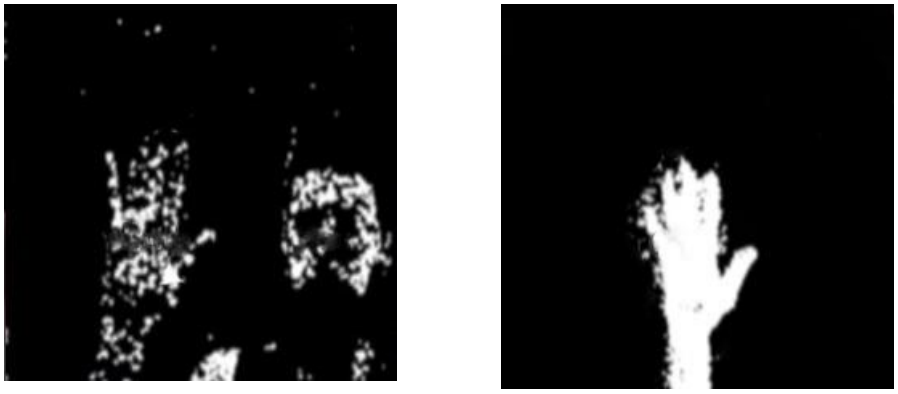


Figure 3-7 Result after apply the algorithm.

C. Feature Extraction and Gesture Training and Recognition:

C.1. Feature Extraction

Once the hand is detected, we apply a feature extraction step. This step (feature extraction) is primordial in our case, since it reduces the hand gesture data which reduces time consumption during the recognition step especially in Smartphones.

There are several feature extraction and data reduction approaches such as HOG, Gabor Wavelet, PCA and LDA. In in our case, we use HOG (SHU et al.2011). since it is fast so the histogram can be calculated quickly.

The basic idea of HOG (Histogram of Oriented Gradient) features is that the local object appearance and shape can often be characterized rather well by the distribution of the local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions. The orientation analysis is robust to lighting changes since the histogram Ming gives translational invariance. The HOG feature summarizes the distribution of measurements within the image regions and is particularly useful for recognition of textured objects with deformable shapes. The method is also simple and fast so the histogram can be calculated quickly. HOG feature is generated for each pixel in the image. The image is divided into several uniformly spaced cells and for each cell a local 1-D histogram of gradient directions or edge orientations is accumulated over all the pixels of the cell. The combined histogram features of all cells form the image representation. The whole process is shown in Fig. 3.4 (SHU et al.2011).

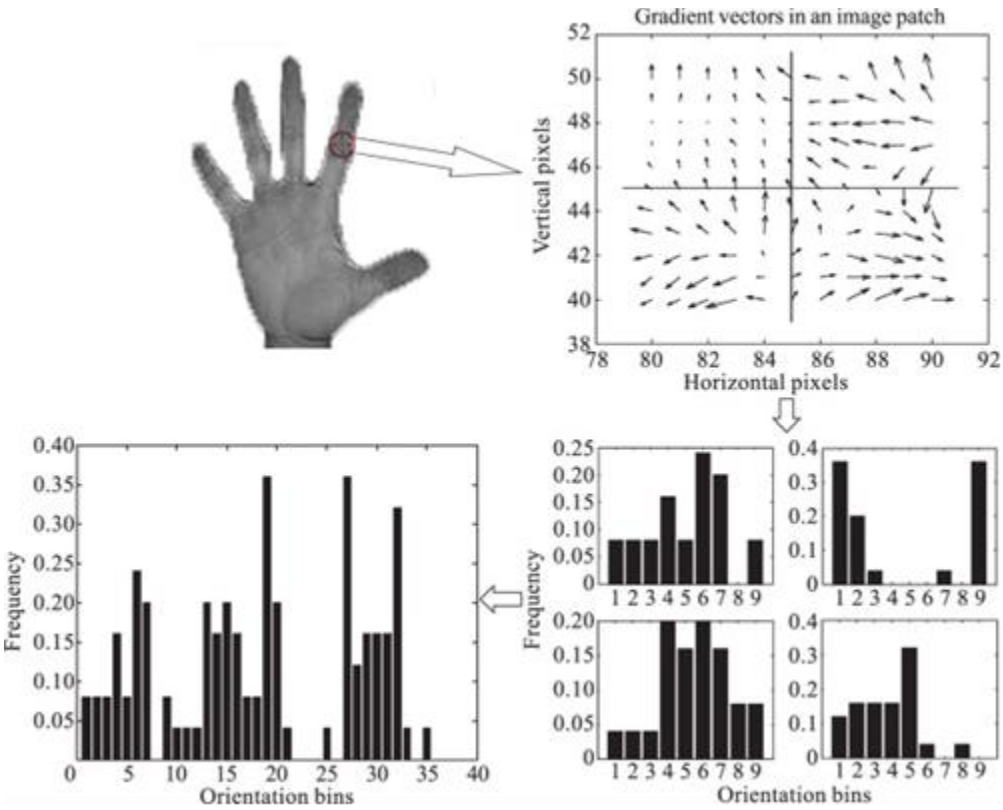


Figure 3-8 Image window divided into small spatial regions ("cells"). Local 1-D histograms of gradient directions or edge orientations are accumulated and concatenated to form the final histogram feature.

Once HOG is applied on hand image, we got a histogram of 144 bins for each image.

The calculated histograms are than used for both training of the neural network in offline, and gesture recognition in real-time steps.

C.2. Neural Network, Training and Recognition

C.2.1 Neural network definition

Artificial neural networks are an old field of research for computer scientists. Works on this technique started as early as 1940, but due to the impossibility to overcome theoretical barriers, they were forgotten during the years 1960-1980. These theoretical barriers have been overcome now, and neural networks are now available for many purposes. A neural network (Richard 1987) is a set of simple units called neurons. A neuron is a linear automata, which realizes a weighted sum of several inputs according to a set of weights, and then computes a Heaviside function or a sigmoid function to obtain an output value, called activation of the neuron. The choice of the transfer function determines whether the neuron is binary or continuously valued. To form a neural network, these neurons are interconnected according to a given topology. This topology defines an input layer, where the activations of the neurons are set to the input values, and an output layer, where the reading of the activations of the neurons gives the answer of the net. Neural networks can therefore be seen as a non-linear transfer function from a vectorial space into another. The input weights of the neurons are the parameters of this transfer function. Presenting an input to the network triggers the computation of the output. A very important property of neural networks is the concept of programming by example. The large number of weights makes it difficult to few them to obtain the desired result. Instead, the net is programmed by example and repetition. It is trained by presenting input-output pairs repeatedly. Each time an input is presented, the network guesses an output. The output part of the input-output pair is used to determine

whether the network is right or wrong. When wrong, the network is corrected by a learning algorithm using a gradient method on the output error to modify the weights. After each modification, the network gets closer to the desired transfer function as represented by the sample base (Debar et al. 1992).

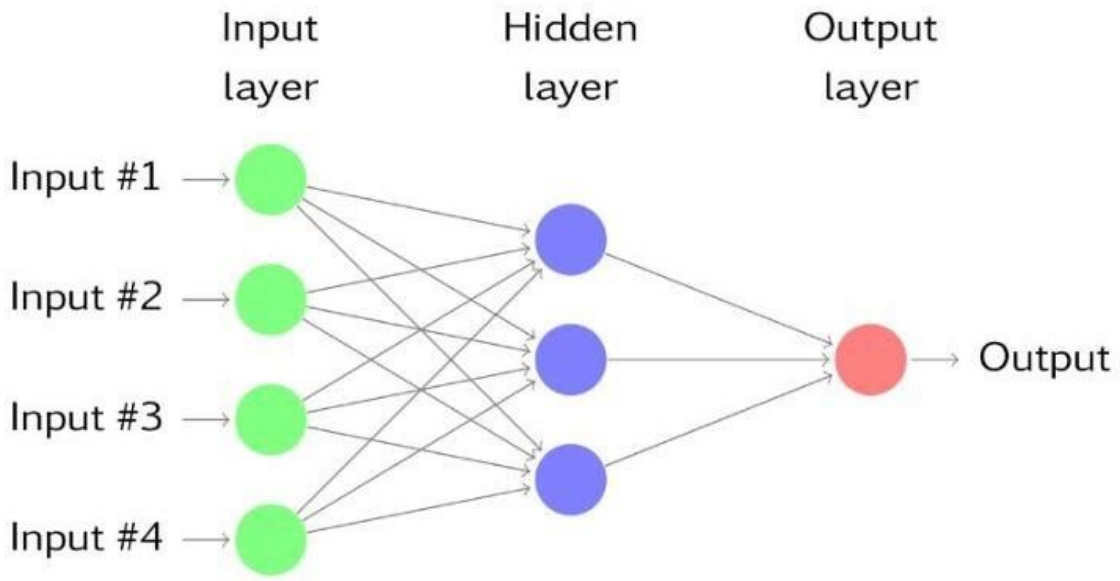


Figure 3-9 A simple neural network

C.2.2 Properties and parameters of neural networks we used

Using the obtained gestures data calculated by HOG, we implemented a neural network with following parameters:

Table 3-1 Parameters of the implemented neural network.

Input data (neuron)	Hidden layer bins	Output (number of defined gestures)
144	**20**	**3**

We used this developed neural network for both training (offline) and testing (real time).For the training step we constructed dataset of 3000 entry of each gesture. The following figure illustrates the recognition process using the developed neural network.

s

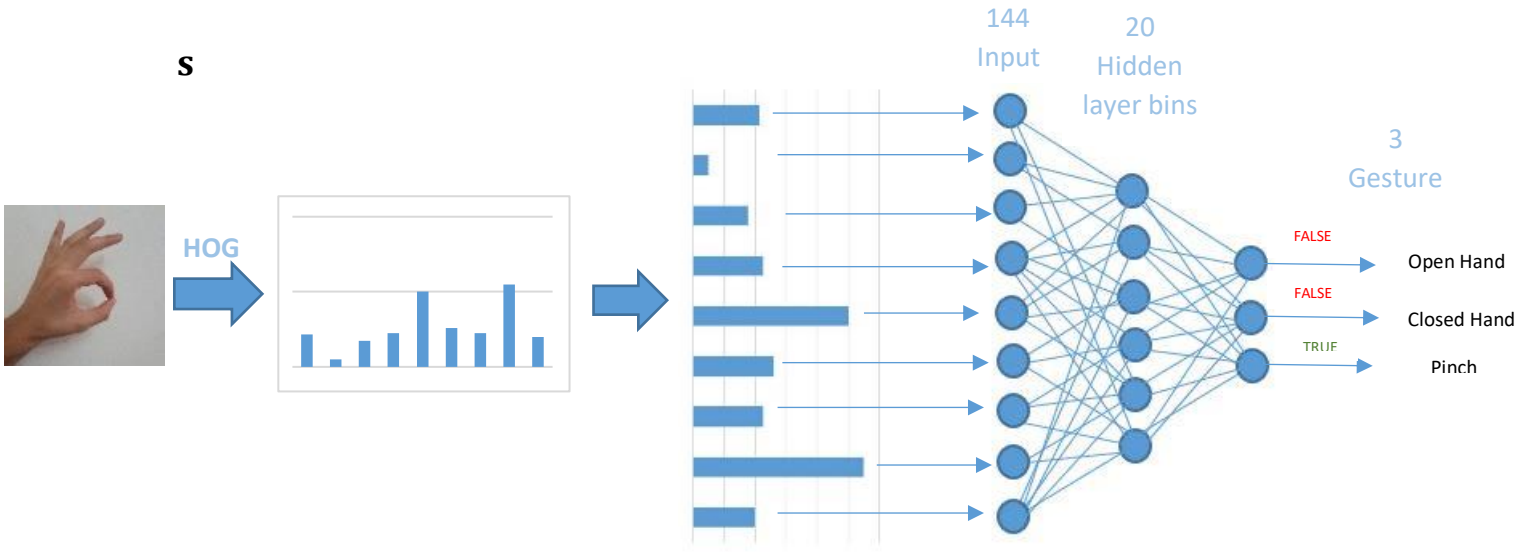


Figure 3-10 Training and Recognition process.

D. 3D interaction:

Once the gestures are recognized we use them to interact with the virtual objects and collaborate with other users.

In order to use the recognized 2D gestures for 3D interaction with virtual objects (rotation –translation), we have to recalculate the 2D hand movements in the virtual 3D space, for this we work with the two axis who facing screen's axis, to apply the manipulation on the 3rd axis we have to make the smartphone screen (camera projection) facing that axis by changing the position of the smartphone. The figures (Figure 3-11, Figure 3-12) illustrate the followed method:

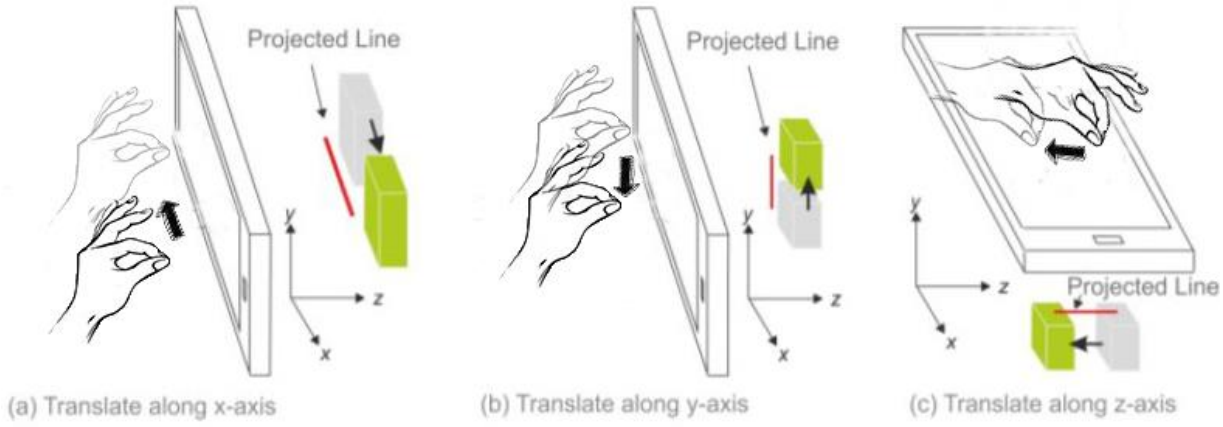


Figure 3-11: Examples of Translations.

Moving the hand up and down in Figure 3-11(a) would cause a translation along the y-axis. Analogously, moving the finger right or left in Figure 3-11(b), respective (c), results in translations along the x-axis.

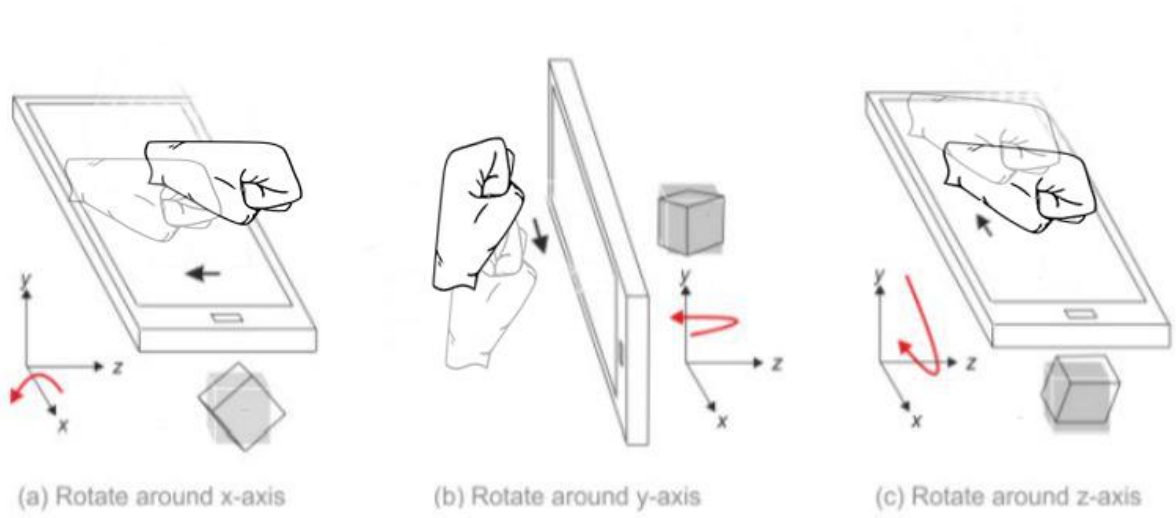


Figure 3-12: Examples of Rotations.

Moving the hand up and down in Figure 3-12(a) would cause a rotation around the x-axis. Analogously, moving the finger right or left in Figure 3-12(c) results in a rotation around the z-axis.

3.4.2 Mixed reality module and collaboration and data sharing module

3.4.2.1 Mixed reality

This part ensures mixed reality module (augmented and virtual reality), therefore there will be two cases:

Augmented reality

In augmented reality case, the 3D pose (position and rotation) of the user has to be determined to be able to add virtual objects in specified locations. In our project we will use Vuforia library for Unity. (for technical detail see chapter 4).

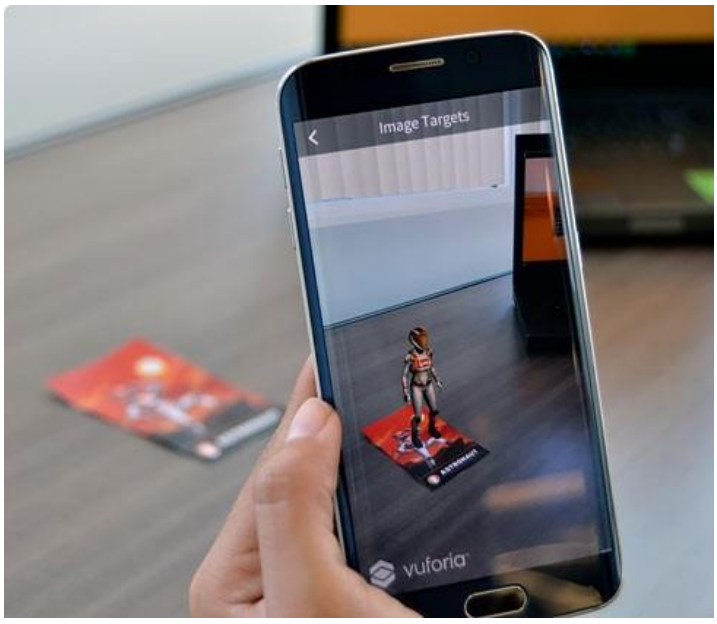


Figure 3-11 Vuforia simple AR display.

3.4.2.2 Collaboration (network):

In parallel with pervious process we send gestures data via the local network using socket. We adopted the Peer to Peer architecture. In this model (Peer to Peer) all the computers of the network are equal, where each one is both a client and a server. Data exchange between two entities is done directly without having to go through a central intermediary.

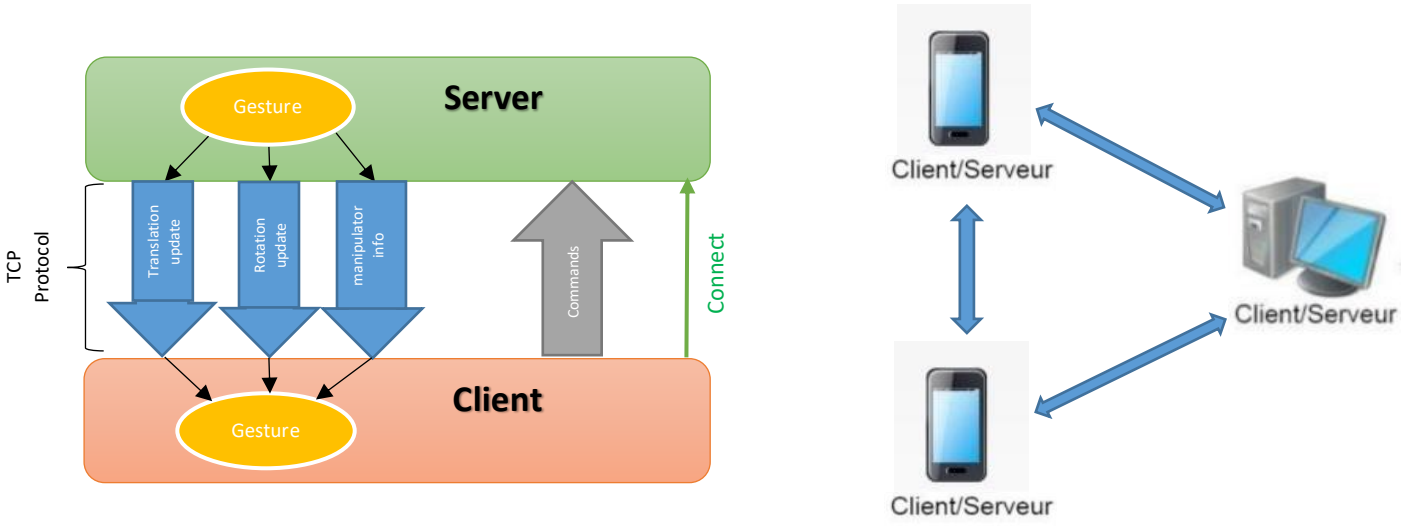


Figure 3-12 Peer to Peer architecture.

Therefore in our project each terminal is a client and a server at the same time, so it can send and receive the recognized gestures and virtual objects data to and from other terminals (users).

3.5 Conclusion

Analysis is a necessary step before designing an application. By exploiting this concept we were able to describe our system under different aspects: technical, functional and behavioral, thus defining the constraints and objectives of the application to be carried out.

We then presented through this chapter the design of our system which is essential during development, which allows us to move on to the implementation which will be the subject of the next chapter.

Chapter 4. Realization and tests

Chapter 4 Implementation and tests

4.1 Introduction

The goal of our master's work is to implement an augmented reality, interactive and collaborative system based on gesture recognition by neural networks. After having presented the design of our system in the previous chapter (chapter 3), we will begin in this chapter the implementation part as well as the scenario on which we based ourselves for the implementation and the tests of our system. This last (system) must allow the collaboration of several users in a 3D environment. Users will need to interact with 3D objects in order to perform different tasks.

4.2 Objectives of the realization

4.2.1 Interaction 3D

As we mentioned in the Introduction above The objective of the application is to allow a user to select and manipulate objects in a Virtual Environment using hand gestures, therefore the goal is to implement a selection and manipulation technology allowing to achieve the desired interaction.

4.2.2 Gestures to recognize

The desired interaction tasks, which are presented in the previous point, require the recognition of a few gestures. Thus with each gesture we associate

a certain action. We give in the following table the correspondence between gestures and actions.

Table 4-1 interaction gestures.

Gesture	Interaction task	Action
Open Hand	Navigation	Navigate in the virtual environment
Closed Hand	Selection /Manipulation	Select a virtual object and rotate it.
Pinch	Selection /Manipulation	Select a virtual object and move it.

We used three postures which are as follows:

a. **Open Hand**: this gesture is recognized when the user opens all these fingers.

b. **Closed Hand:** this gesture is recognized when the user closes all these fingers.

c. **Pinch:** this gesture is recognized when the user joins their thumb to their index finger while closing the rest of these fingers.

The following figure illustrates the gestures in pictures.

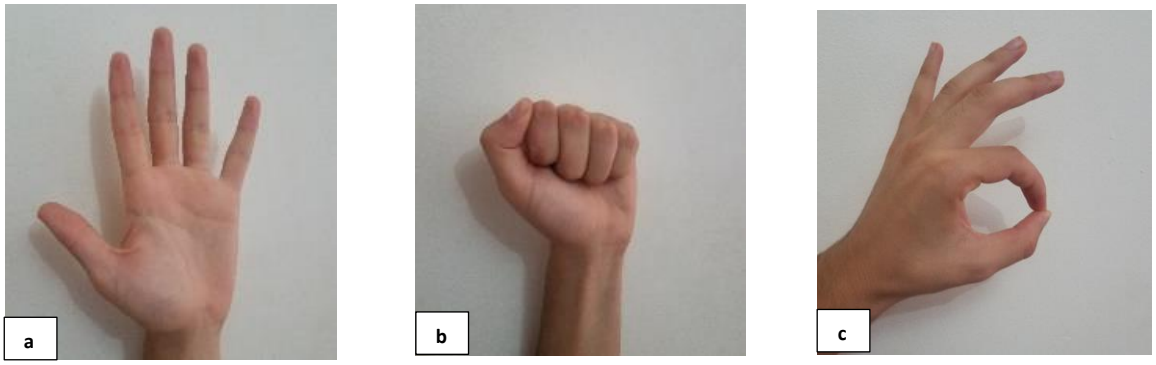


Figure 4-1 Images of gestures used for 3D interaction.

4.3 Software environment

It consists of a set of software tools for the development and implementation of our system. It's about:

- ✓ Game engine Unity 2019.3.0f6.
- ✓ OpenCV for Unity.
- ✓ Vuforia augmented reality SDK.
- ✓ Visual Studio (C # programming language).

4.3.1 Unity 2019.3.0f6

Unity 2019.3.0f6 is a platform for developing 3D video games. It is a tool that integrates all the features needed to make an animated 3D scene like video games. In Unity 2019.3.0f6, the program is driven by scripts that can be expressed either in C # or JavaScript.

Unity 2019.3.0f6 also has a rich and easily customizable graphical interface (shown in figure 4-1). It allows you to work on the game interactively, this interface allows for example to:

- ✓ Place the objects in 3D space.
- ✓ Apply geometric transformations to objects in the scene using the mouse.
- ✓ Associate animations with different 3D objects.
- ✓ Modify textures, materials, physical properties, collision parameter, etc.

Unity's graphical interface is divide into five panels (figure 4-2):

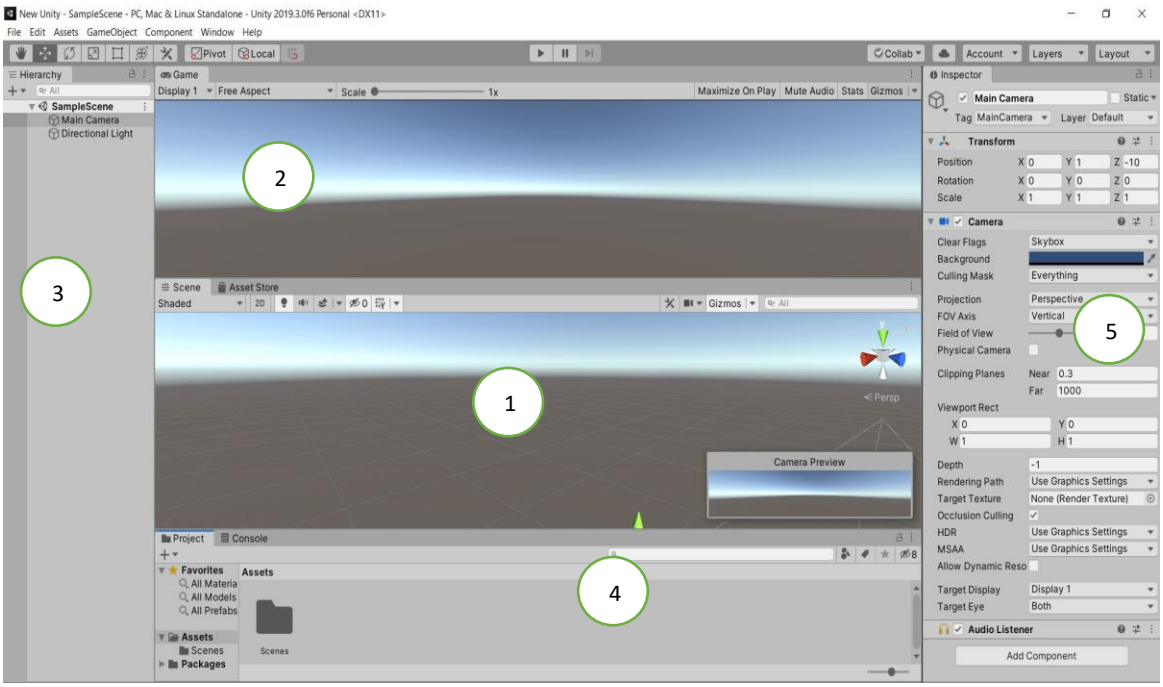


Figure 4-2 Unity interface.

1. Scene: It is in the Scene tab that we can position Game Objects, edit them and manipulate them. This window offers different means of control allowing you to vary the level of detail. These controls can be used to turn lights on and off, to display objects with or without their textures, to modify the coloring of the scene (RGB, Alpha, Overdraw), or to change the viewing angle in the scene.

2. **Game:** The game window displays the result of the execution while the scene is being edited and therefore allows a preview of the game or the virtual environment. This tab can be maximized in order to preview in full screen mode.

3. **Hierarchy:** The Hierarchy panel contains a list of all Game Objects contained by the Stage, so it is just the Stage under another view. A Game Object can be tangible such as a cube or sphere, or intangible such as a camera or a light source. In Unity 3D, Game Objects are organized in a hierarchy, thereby defining the notion of "parent / child". Thus a Game Object can contain several descendants in a hierarchy, and any transformation applied to a parent is inherited by all its descendants, on the other hand the transformations applied to a Game Object are not transmitted to its parent.

4. **Project:** The Project tab displays the contents of a special folder called "Assets". Unity automatically creates this folder when creating a project. The latter lists all the elements (files, images, scripts, etc.) that are used to create Game Objects in a project and thus design a game or a virtual environment.

5. **Inspector:** The Inspector tab is a pop-up panel, which means that it changes depending on what's selected elsewhere in Unity. It is in this tab that we can adjust the position, rotation and scale of a Game Object present in the hierarchy tab. The Inspector panel can also display controls for adding, removing, and configuring components that add functionality to Game Objects.

4.3.2 OpenCV

Open Source Computer Vision Library, OpenCV for short, is a cross-platform library computer vision based on open source distribution. It can run on Linux, Windows and Mac OS operating system. It's lightweight and efficient-makes up of a series of C function and a small amount of C + + class, at the same time provides the Python, Ruby, MATLAB language interface; realize the many general algorithm of image processing and computer vision. Very rich visual processing algorithm provided by the OpenCV, with C language to write and it part, combined with the characteristics of its open source, it does not need to add new external support, which can also complete the compilation of links to generate executable program. As a result, many people use it to do transplant of the algorithm. Whether still images and video sequence images, using OpenCV function and classifier are relatively easy to face detection, the process is shown in Figure 4-3 (SIGUT et al.2020).

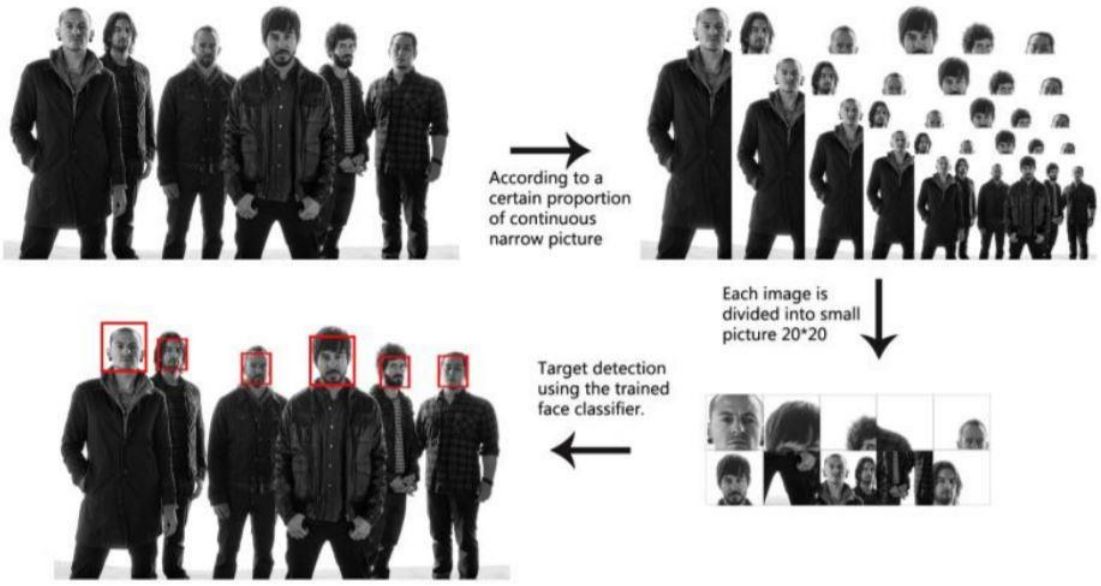


Figure 4-3 The Flow Chart of OpenCV Face Detection.

4.3.3 Vuforia

Vuforia is an AR SDK for smartphones or other similar mobile device that allows executes AR applications into a real time video obtained from these devices. This software uses the capabilities of the computer vision technology to recognize and make the individually tracking of the objects captured by the video camera in real time.

The capability of Vuforia to image registration enables developers to position and orient in the space the virtual objects, mainly 3D objects or other type of media, in relation to real world images or video when these are viewed through the camera of a smartphone. The virtual object then can track the position and orientation of the real image in real time, so that the viewer's perspective on the object corresponds with their perspective on the real world target. In this way, the virtual object or objects appears as if from another real world object.

The Vuforia SDK supports different types of targets, both 2D and 3D, including multi-target configurations, marker less image targets and frame markers. The SDK have another additional features like localized occlusion detections using virtual buttons, image target selection in real time and the ability to reconfigure and create target sets depending on the scenario.

But these not only will be made with Vuforia SDK to create AR applications. In fact, Vuforia provides an API6 in Java, C++ and .Net languages through an extension to the Unity3D game engine. Unity is an integrated authoring tool for creating 3D video games or other interactive content such as architectural visualizations or real-time 3D animations. Unity consists of both an editor for developing/designing content and a game engine for executing the final product (SIMONETTI et al.2013).

Development Environment	Development Platform			
	Native SDK		Unity Extension	
	Android	iOS	Android	iOS
Windows	Yes	--	Yes, multi-platform deployment	
MacOS	Yes	Yes	Yes, multi-platform deployment	
Linux	Yes	--	--	--

Figure 4-4 Vuforia development platforms.

4.3.4 Visual Studio (C # programming language)

Microsoft Visual Studio (Microsoft, 2019), is a set of development software for Windows designed by Microsoft, offering the ability to generate different

types of applications. This development environment is used by many programming languages such as Basic, C ++, C # and many others.

A multitude of versions exist since Visual Studio 1997. We have used in our work Visual Studio 2019, which offers many tools to facilitate development and to generate quality solutions.

As for the programming language, we used the C Sharp (C #) language, a simple, rich and powerful object-oriented programming language, also created by Microsoft (Microsoft 2019).

4.4 Implementation

As we mentioned previously our project is composed from two parts (augmented reality and virtual reality), for this we have implemented two scenes where each one allows the user to share and to visualize virtual objects in a specific part. Also, he can interact in 3D and co-manipulate shared objects thanks to a network manager using gesture recognition.

4.4.1 Test of the gesture recognition module

We implemented the gesture recognition module in c# + unity and we built it for two platforms (Windows and Android), after several tests we got following results of the recognition rate (table 4-2 and table 4-3):

Table 4-2 cross test for recognition rate on Windows.

	Open Hand	Closed Hand	Pinch
Open Hand	**0.91**	**0.01**	**0.07**
Closed Hand	**0.02**	**0.83**	**0.14**
Pinch	**0.06**	**0.08**	**0.87**

Table 4-3 cross test for recognition rate on Android.

	Open Hand	Closed Hand	Pinch
Open Hand	**0.89**	**0.07**	**0.10**
Closed Hand	**0.04**	**0.77**	**0.09**
Pinch	**0.06**	**0.15**	**0.81**

We calculated also the mean calculation time for every frame for both windows and Android see (table 4-4).

Table 4-4 the mean calculation time for every frame.

	Windows	Android
Time (ms)	**43.4**	**90.9**
Fps	**23**	**11**

4.4.2 Augmented reality scene

In this scene the user can visualize virtual objects by augmented reality, and interact (navigate, select and manipulate) in 3D and co-manipulate shared objects using gesture recognition. In order to perform the interaction tasks we mentioned previously we have in our application three modes, each one represents a task of 3D interaction. We will explain each one separately in the following:

A. Navigation mode: in this mode the user can navigate in the augmented reality scene thanks to the virtual hand. The 2D movements of the virtual hand are related to the user hand performing the "Open Hand" gesture (see chapter

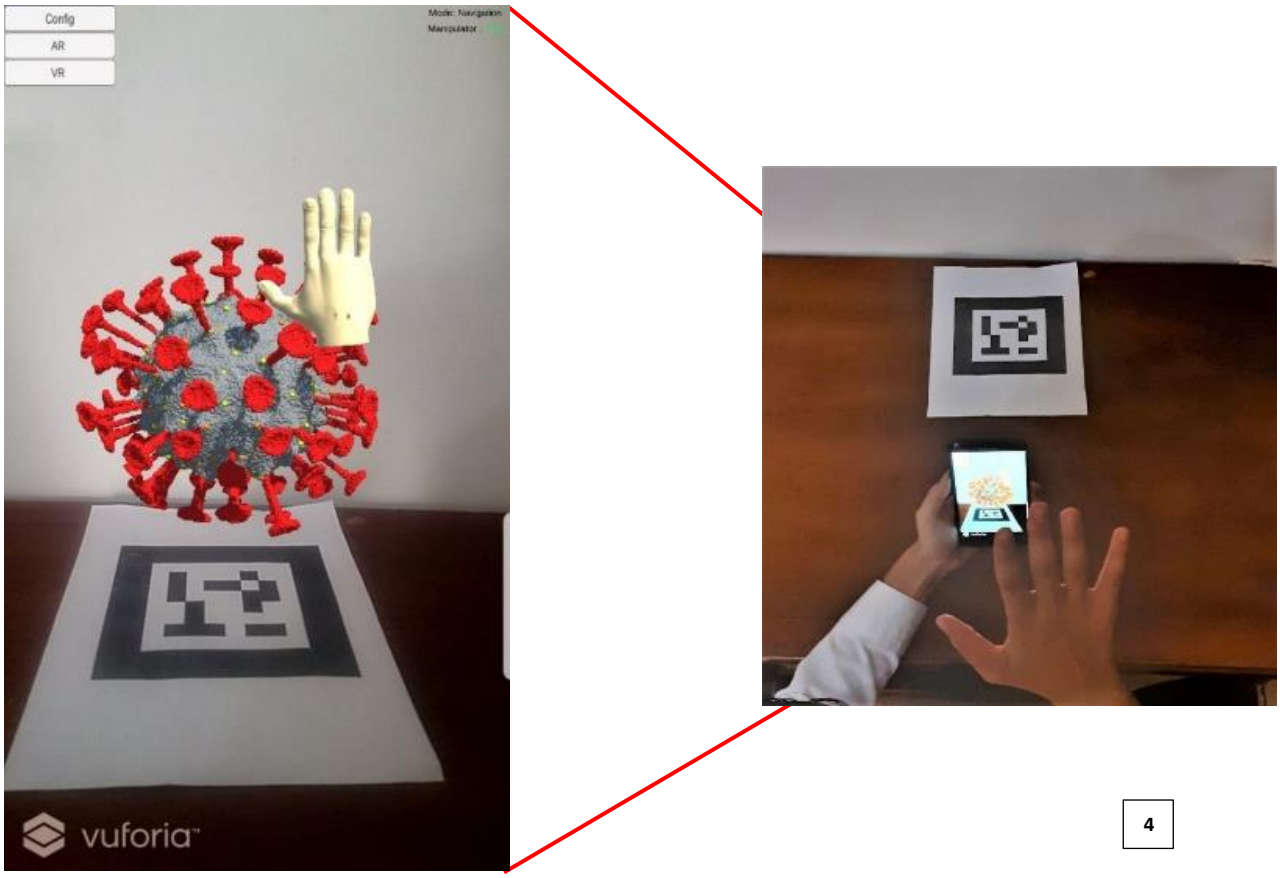


Figure 4-5 Navigation mode.

1) **Vuforia Marker.**

2) **Virtual object.**

3) **Virtual hand on open hand gesture following the user's hand.**

4) **The user acting open hand gesture.**

B. Rotation mode: after the selection of the virtual object, the user can rotate it in this mode. Rotation of the virtual object is related to the movement of the user's hand performing the "Close Hand" gesture (see chapter 4 section 4.2.2), up and down would cause a rotation around the horizontal facing axis right and left for the rotation around the vertical facing axis.

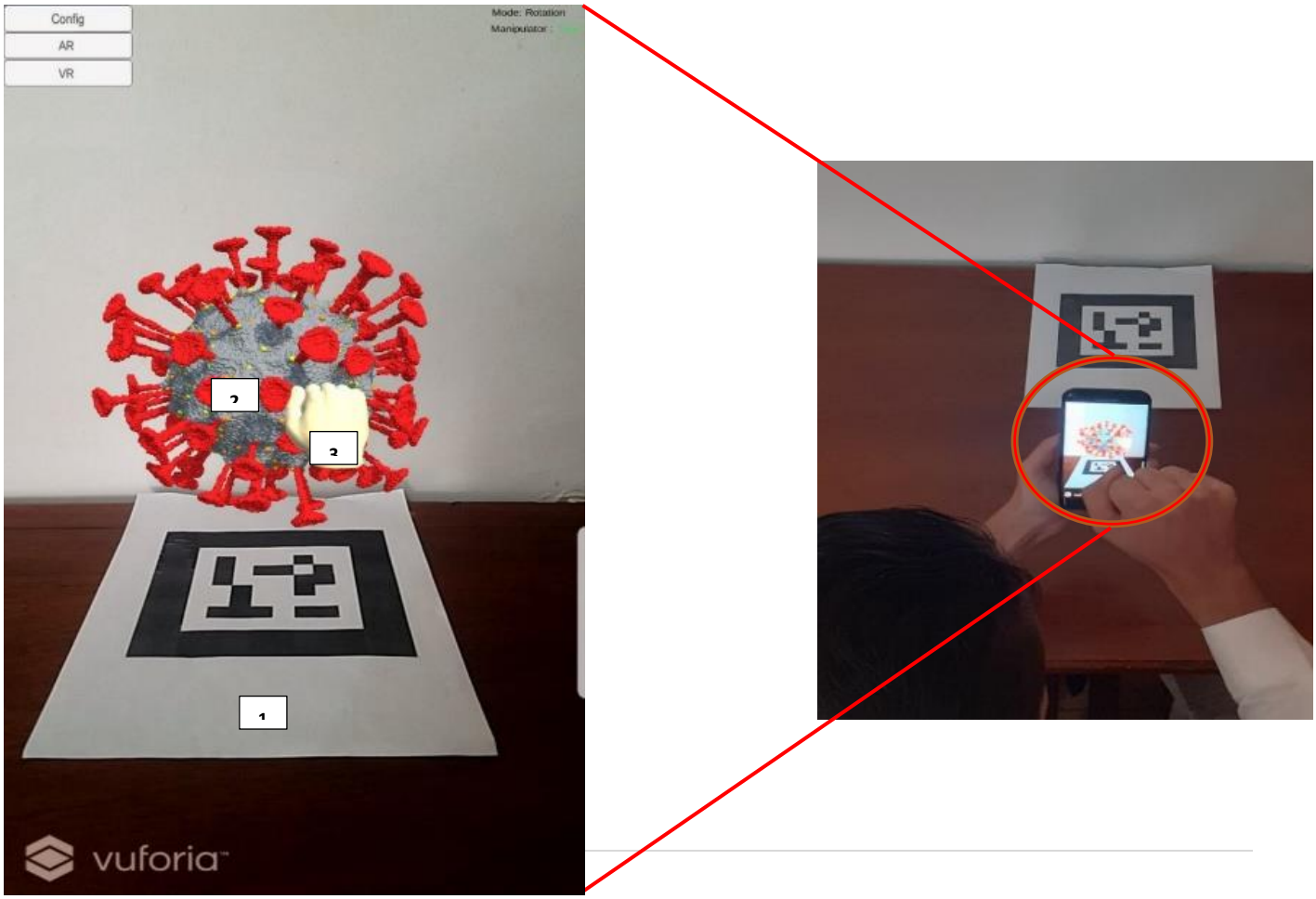


Figure 4-6 Rotation mode.

1) **Vuforia Marker.**

2) **Virtual object.**

3) **Virtual hand on open hand gesture following the user's hand.**

4) **The user acting closed hand gesture.**

C. Translation mode: in this mode the user can move virtual objects in the environment after they being selected. In order to move the virtual object the user must do the pinch gesture. Once the gesture are recognized, the movements of the selected virtual object would be related to the movements of the user's virtual hand performing "Pinch" gesture (see chapter 4 section 4.2.2).

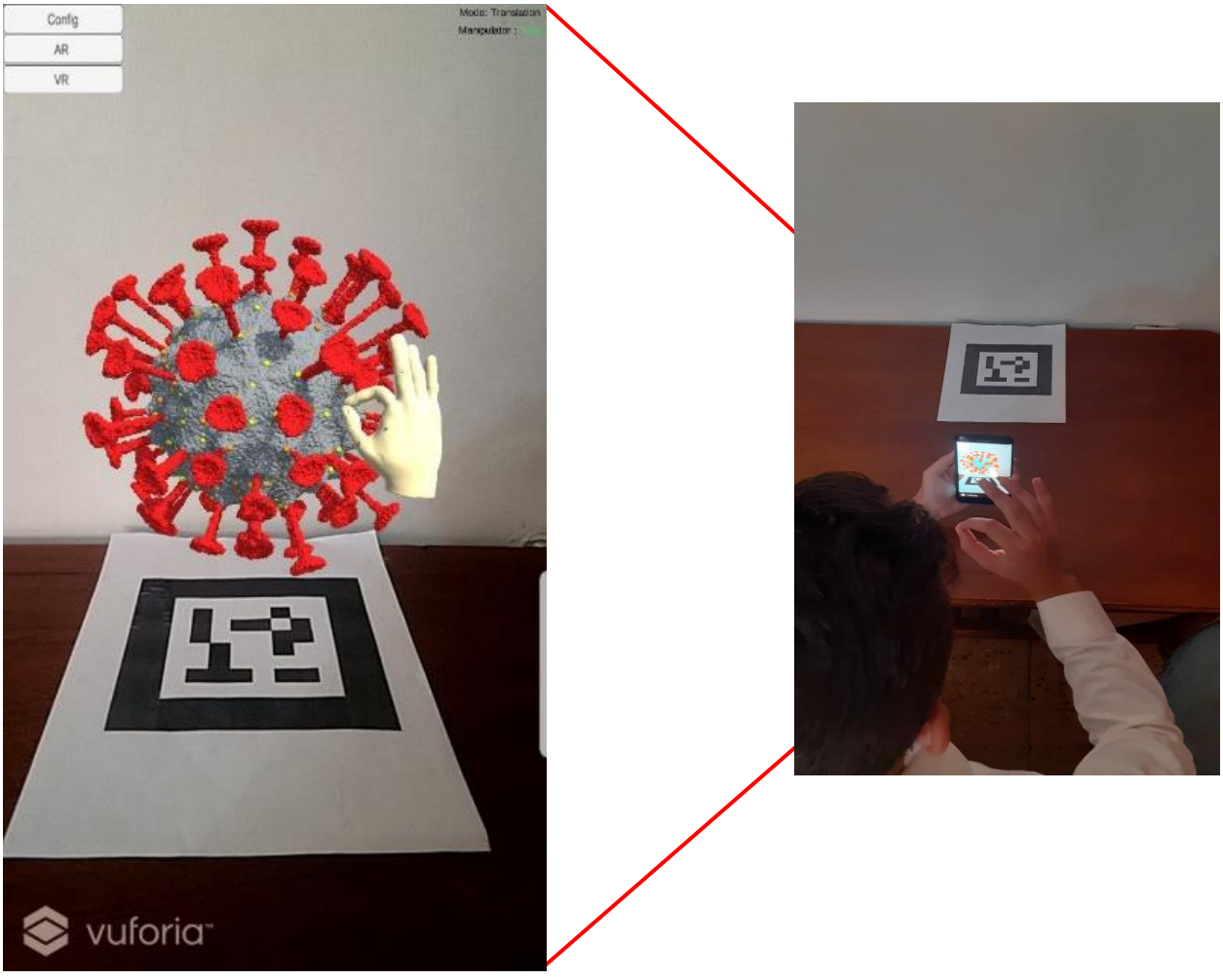


Figure 4-7: Translation mode.

1) **Vuforia Marker.**

2) **Virtual object.**

3) **Virtual hand on open hand gesture following the user's hand.**

4) **The user acting Pinch gesture.**

Within the development of the application we found that some smartphone can't use front and back cameras in the same time, so we developed the second part (virtual reality part) to allow the users of those smartphone to collaborate with other users (augmented reality users).

4.4.3 Virtual reality scene

Same as the augmented reality, the user can interact in 3D and co-manipulate shared objects using gesture recognition, in this part the user visualize virtual objects by virtual reality. The following figure illustrate the virtual reality scene with the different interaction tasks.

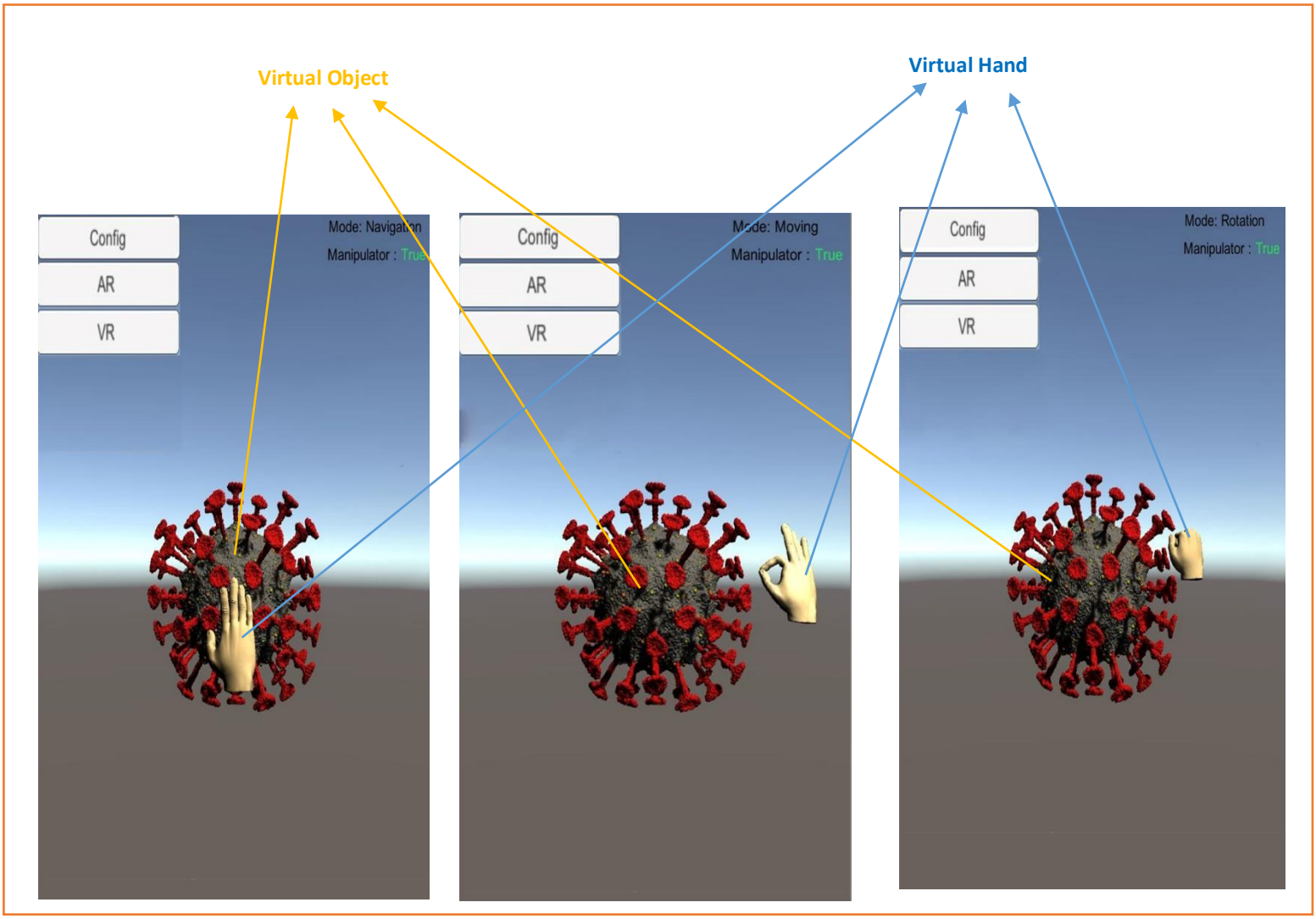


Figure 4-7 Virtual reality scene

4.4.4 Network manager

In order to make the users collaborate and co-manipulate shared virtual objects we developed a network manager to manage the users connection and disconnection (add and delete), the following figure illustrates the interface of the network manager

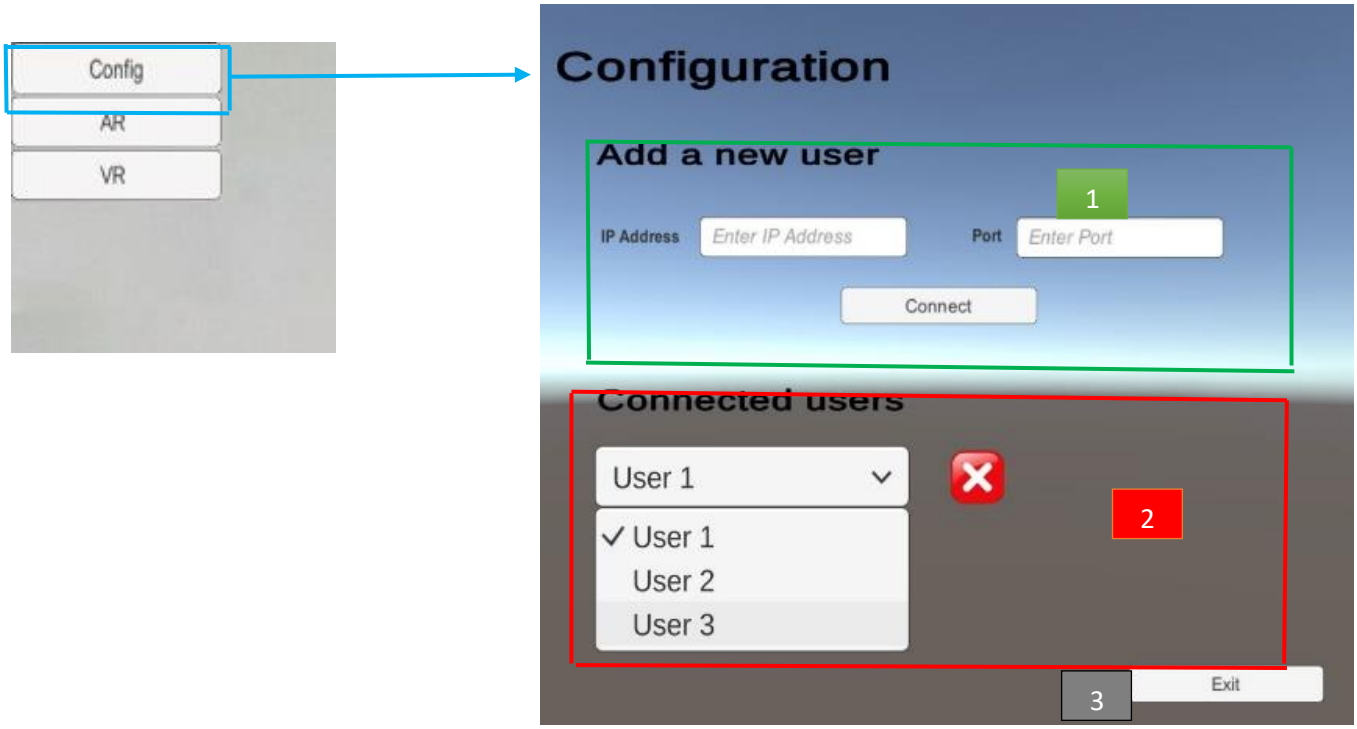


Figure 4-8 Network manager.

1. **Add a new user: adding by his IP address and port**
2. **Connected users: is the list of the users connected to the network we can delete it after being selected by the button .**
3. **Exit: which allows you to close the interface.**

4.2 Conclusion

We presented through this last chapter the tools that we used for the realization of our work, as well as the detailed implementation that we followed in order to arrive at a functional application.

GENERAL CONCLUSION

Virtual reality (VR) and augmented reality (AR) are two technologies that have imposed themselves on topical research thanks to their many applications which affect different fields: medicine, industry, education, marketing and many others. AR by its ability to realistically embed virtual objects in a real scene and VR by the possibility of immersing the user in a virtual environment allow to realize very interesting applications.

The bibliographical research that we carried out allowed us to observe that the two paradigms (AR, VR) have also revolutionized the field of interaction. Indeed, the notion of interaction is now used to designate a set of rules and techniques allowing the user to perform interaction tasks within a virtual or augmented environment.

The objective of our work was to implement a technique allows to interact in 3D and co-manipulate shared objects in a virtual or augmented environment using gesture recognition by neural network.

The result of our efforts is a 3D representation of the hand reproducing the user's gestures in real time, functioning in both augmented reality and virtual reality. However, the mainly implemented interaction tasks are selection and manipulation, Also our project is a cross platform application (Windows, Android).

In conclusion, we would like to emphasize that this work has been a pleasant experience.

REFERENCES

(Richard 1987)	*An Introduction to computing with neural nets, Richard P. Lippmann, IEEE ASSP Magazine, April 1987, pages 4-22.*
(Howard 1991)	*Howard Rheingold (1991). Virtual Reality. Summit Books, London.*
(Cootes et Taylor 1992)	*Cootes TF, Taylor CJ (1992) Active shape models smart snakes. In: British machine vision conference, pp 266–275*
(Debar et al. 1992)	*DEBAR, Herve, BECKER, Monique, et SIBONI, Didier. A neural network component for an intrusion detection system. In: null. IEEE, 1992. p. 240.*
(Held et Durlach 1993)	*R. Held, N. Durlach: Telepresence. Presence, Vol. 1, No. 1, pp. 109-113 (1993).*
(Coyne & Heidegger 1994)	*Coyne, R., Heidegger and virtual reality: The implications of Heidegger's thinking for computer representations. Leonardo: Journal of the International Society for Arts, Sciences, and Technology, 27 (1994).*
(Milgram et al. 1994)	*Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (1994). Augmented reality: a class of displays on the reality-virtuality continuum. In Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series (Vol. 2351, pp. 282-292). 5*
(Rehg & Kanade 1994)	*20 J. M. Rehg and T. Kanade, "DigitEyes: Vision-based hand tracking for human-computer interaction," in Proc. IEEE Workshop Motion NonRigid Agricult. Objects, Austin, TX, USA, 1994, pp. 16–22*
(Crowley et al. 1995)	*Crowley J, Berard F, Coutaz J (1995) Finger tracking as an input device for augmented reality. In: International workshop on gesture and face recognition, Zurich*
(Lee & Kunii 1995)	*Lee J, Kunii TL (1995) Model-based analysis of hand posture. IEEE Comput Graphics Appl 15(5):77–86 Lee D, Park Y (2009) Vision-based remote control system by motion detection and open finger counting. IEEE Trans Consumer Electron 55(4):2308–2313*
(Yuan et al. 1995)	*Yuan Q, Sclaroff S, Athitsos V (1995) Automatic 2D hand tracking in video sequences. In: IEEE workshop on applications of computer vision, pp 250–256*
(Cui et Weng 1996)	*Cui Y, Weng J (1996) Hand sign recognition from intensity image sequences with complex background. In: Proceedings of the IEEE computer vision and pattern recognition (CVPR), pp 88–93*
(Darrell et al. 1996)	*Darrell T, Essa I, Pentland A (1996) Task-specific gesture analysis in real-time using interpolated views. IEEE Trans Pattern Anal Mach Intell 18(12):1236–1242*
(Kalawsky 1996)	*Kalawsky, R.S. AGOCG Report. Exploiting Virtual Reality Techniques in Education and Training: Technological Issues (1996) [3] Coyne, R., Heidegger and virtual*
(Rekimoto 1996)	*Rekimoto, J. Transvision: A Hand-held Augmented Reality System for Collaborative Design. In Proceeding of Virtual Systems and Multimedia '96 (VSMM '96), Gifu, Japan, 18-20 Sept., 1996.*
(Schmalsteig et al. 1996)	*Schmalsteig, D., Fuhrmann, A., Szalavari, Z., Gervautz, M., Studierstube - An Environment for Collaboration in Augmented Reality. In CVE '96 Workshop Proceedings, 19-20th September 1996, Nottingham, Great Britain.*

(Isard & Blake 1998)	*Isard M, Blake A (1998) Condensation—conditional density propagation for visual tracking. Int J Comput Vis 29(1):5–28*
(Martin et al. 1998)	*Martin J, Devin V, Crowley J (1998) Active hand tracking. In: IEEE conference on automatic face and gesture recognition, Nara, Japan, pp 573–578*
(Bowman 1999)	*D. A. Bowman, « Interaction Techniques for Common Tasks in Immersive Virtual Environments », These de doctorate, Georgia Institute of Technology, 1999. 9*
(Hirokazu & Billinghurst 1999)	*K. Hirokazu et M. Billinghurst, « Marker suivi and HMD calibration for a video-based augmented reality conferencing system », Proc. 2nd IEEE ACM Int. Work. Augment. Real., p. 85-94, 1999.*
(Terrillon et al. 2000)	*Terrillon J, Shirazi M, Fukamachi H, Akamatsu S (2000) Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In: Proceedings of the international conference on automatic face and gesture recognition (FG), pp 54–61*
(Wu & Huang 2000)	*Wu Y, Huang TS (2000) View-independent recognition of hand postures. In: Proceedings of the IEEE computer vision and pattern recognition (CVPR), vol 2. Hilton Head Island, SC, pp 84–94*
(Mammen et al. 2001)	*Mammen JP, Chaudhuri S, Agrawal T (2001) Simultaneous tracking of both hands by estimation of erroneous observations. In: Proceedings of the British machine vision conference (BMVC), Manchester*
(Viola & Jones 2001)	*Viola P, Jones M (2001) Robust real-time object detection. In: IEEE workshop on statistical and computational theories of vision, Vancouver*
(Schapire 2002)	*Schapire R (2002) The boosting approach to machine learning: an overview. In: MSRI workshop on nonlinear estimation and classification*
(Argyros et Lourakis 2004)	*Argyros A, Lourakis MIA (2004) 3D tracking of skin-colored regions by a moving stereoscopic observer. Appl Opt 43(2):366–378*
(Fiala 2004)	*M. Fiala, « Artag revision 1, a fiducial marker system using digital techniques », 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., no November, p. 590-596, 2004.*
(Sigal et al. 2004)	*Sigal L, Sclaroff S, Athitsos V (2004) Skin color-based video segmentation under time-varying illumination. IEEE Trans Pattern Anal Mach Intell 26(7):862–877*
(Henrysson et al. 2005a)	*A. Henrysson, M. Billinghurst, and M. Ollila, "Face to face collaborative AR on mobile phones," in Proc. 4th IEEE ACM Int. Symp. Mixed Augmented Reality, Vienna, Austria, Oct. 2005, pp. 80–89.*
(Henrysson et al. 2005b)	*A. Henrysson, M. Billinghurst, and M. Ollila, "Virtual object manipulation using a mobile phone," in Proc. Int. Conf. Augmented Tele-Existence Christchurch (ICAT), New York, NY, USA, 2005, pp. 164–171*
(Henrysson et al. 2005c)	*A. Henrysson, M. Billinghurst, and M. Ollila, "Mobile phone based AR scene assembly," in Proc. 4th Int. Conf. Mobile Ubiquitous Multimedia, Christchurch, New Zealand, 2005, pp. 95–102*
(Cote et al. 2006)	*Cote M, Payeur P, Comeau G (2006) Comparative study of adaptive segmentation techniques for gesture analysis in unconstrained environments. In: IEEE international workshop on imagining systems and techniques, pp 28–33*

(Yilmaz et al. 2006)	*Yilmaz JA, Javed O, Shah M (2006) Object tracking: a survey. ACM Comput Surv 38:13 Yin M, Xie X (2003) Estimation of the fundamental matrix from uncalibrated stereo hand images for 3D hand gesture recognition. Pattern Recogn 36(3):567–584*
(TEICHRIEB et al. 2007)	*TEICHRIEB, Veronica, LIMA, J. P. S. M., APOLINÁRIO, E., et al. A survey of online monocular markerless augmented reality. International Journal of Modeling and Simulation for the Petroleum Industry, 2007, vol. 1, no 1, p. 1-7.*
(Bradski et Kaehler 2008)	*Bradski G, Kaehler A (2008) Learning OpenCV, O'Reilly, pp 337–341*
(Jinjun et al. 2009)	*R. Jinjun, G. Tongyue, G. Zhenbang, et J. Zhen, « Low Cost Hand Gesture Learning and Recognition System Based on Hidden Markov Model », in Proceedings of the Second International Symposium on Information Science and Engineering, 2009, p. 433-438.*
(Kan et al. 2009)	*T.-W. Kan, C.-H. Teng, et W.-S. Chou, « Applying QR code in augmented reality applications », Proc. 8th Int. Conf. Virtual Real. Contin. its Appl. Ind. VRCAI 09, vol. 1, no 212, p. 253, 2009.*
(Stergiopoulou & Papamarkos 2009)	*E. Stergiopoulou et N. Papamarkos, « Hand gesture recognition using a neural network shape fitting technique », Eng. Appl. Artif. Intell., vol. 22, no 8, p. 1141-1158, 2009.*
(Wang & Popović 2009)	*R. Y. Wang et J. Popović, « Real-time hand- tracking with a color glove », ACM Trans. Graph., vol. 28, no 3, p. 1, 2009.*
(Zabulis et al. 2009)	*Zabulis X, Baltzakis H, Argyros A (2009) Vision-based Hand gesture recognition for human–computer interaction. In: The Universal Access Handbook. LEA*
(Atchesonet et al. 2010)	*B. Atcheson, F. Heide, et W. Heidrich, « CALTag: High Precision Fiducial Markers for Caméra Calibration », in Vision, Modeling, and Visualization, 2010.*
(BELLARBI et al. 2011)	*BELLARBI, Abdelkader, BENBELKACEM, Samir, ZENATI-HENDA, Nadia, et al. Hand gesture interaction using color-based method for tabletop interfaces. In : 2011 IEEE 7th International Symposium on Intelligent Signal Processing. IEEE, 2011. p. 1-6.*
(SHU et al. 2011)	*SHU, Chang, DING, Xiaoqing, et FANG, Chi. Histogram of the oriented gradient for face recognition. Tsinghua Science and Technology, 2011, vol. 16, no 2, p. 216-224.*
(Bellarbi et al. 2011)	*A. Bellarbi, S. Benbelkacem, N. Zenati-Henda, et M. Belhocine, « Hand gesture interaction using color-based method for tabletop interfaces », in WISP 2011 - IEEE International Symposium on Intelligent Signal Processing, Proceedings, 2011, p. 180-185.*
(Ding et al. 2011)	*Y. Ding, H. Pang, X. Wu, et J. Lan, « Recognition of hand-gestures using improved local binary pattern », in 2011 International Conference on Multimedia Technology, ICMT 2011, 2011, p. 3171-3174.*
(Fahn & Chu 2011)	*C.-S. Fahn et K.-Y. Chu, « Hidden-Markov-model-based hand gesture recognition techniques used for a human-robot interaction system », in 14th International Conference on Human-Computer Interaction, HCI International 2011, 2011, vol. 6762 LNCS, no PART 2, p. 248-258.*
(Huang et al. 2011)	*D. Y. Huang, W. C. Hu, et S. H. Chang, « Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination », Expert Syst. Appl., vol. 38, no 5, p. 6031-6042, 2011.*

(Morris 2011)	*J. Morris, Android User Interface Development: Beginner's Guide. Birmingham, U.K.: Packt, 2011*
(Wang et al. 2011)	*R. Y. Wang, S. Paris, et J. Popovi, « 6D Hands: Markerless Hand tracking for Computer Aided Design », ACM User Interface Softw. Technol., 2011.*
(Benbelkacem et al. 2012)	*Benbelkacem, S., Zenati-Henda, N., Belhocine, M., Bellarbi, A. and Tadjine, M., 2012, May. Interactive space for management of documents in a maintenance context. In Multimedia Computing and Systems (ICMCS), 2012 International Conference on (pp. 378-383). IEEE*
(Holz et al. 2012)	*D. Holz, K. Hay, et M. Buckwald, « Electronic sensor », US Patent., 2012.*
(Liu et al. 2012)	*2 Y. Liu, Y. Yin, et S. Zhang, « Hand Gesture Recognition Based on Hu Moments in Interaction of Virtual Reality », in International Conference on Intelligent Human-Machine Systems and Cybernetics, 2012, p. 145-148.*
(Rautaray & Agrawal 2012)	*S. S. Rautaray et A. Agrawal, « Vision based hand gesture recognition for human computer interaction: a survey », Artif. Intell. Rev., vol. 43, no 1, 2012.*
(SIMONETTI et al. 2013)	*SIMONETTI IBAÑEZ, Alexandro et PAREDES FIGUERAS, Josep. Vuforia v1. 5 SDK: Analysis and evaluation of capabilities. 2013. Thèse de maîtrise. Universitat Politècnica de Catalunya.*
(Bellarbi et al. 2013)	*A. Bellarbi, H. Belghit, S. Benbelkacem, N. Zenati, et M. Belhocine, « Hand gesture recognition using contour based method for tabletop surfaces », in 10th IEEE International Conference on Networking, Sensing and Control, p. 832-836, 2013,*
(Lin et al. 2013)	*W.-S. Lin, Y.-L. Wu, W.-C. Hung, et C.-Y. Tang, « A Study of Real-Time Hand Gesture Recognition Using SIFT on Binary Images », Smart Innovation, Systems and Technologies, vol. 21. Springer Berlin Heidelberg, p. 235-246, 2013.*
(Melax et al. 2013)	*S. Melax, L. Keselman, et S. Orsten, « Dynamics based 3D skeletal hand tracking », Graph. Interface Conf., p. 63-70, 2013.*
(Priyal & Bora 2013)	*S. P. Priyal et P. K. Bora, « A robust static hand gesture recognition system using geometry based normalizations and Krawtchouk moments », Pattern Recognit., vol. 46, no 8, p. 2202-2219, 2013.*
(Yao & Li 2013)	*Y. Yao et C. T. Li, « Real-time hand gesture recognition for uncontrolled environments using adaptive SURF tracking and hidden conditional random fields », in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013, vol. 8034 LNCS, no PART 2, p. 542-551.*
(Hasan et Abdul-Kareem 2014)	*H. Hasan et S. Abdul-Kareem, « Static hand gesture recognition using neural networks », Artif. Intell. Rev., vol. 41, no 2, p. 147-181, févr. 2014.*
(Qian et al. 2014)	*C. Qian, X. Sun, Y. Wei, X. Tang, et J. Sun, « Realtime and robust hand tracking from depth », in Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, 2014, p. 1106-1113.*
(Tompson et al. 2014)	*J. Tompson, K. Perlin, Y. LeCun, et M. Stein, « Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks », ACM Trans. Graph., vol. 33, no 5, p. 110-169, 2014.*

(Khamis et al. 2015)	*S. Khamis, J. Taylor, J. Shotton, C. Keskin, S. Izadi, et A. Fitzgibbon, « Learning an efficient model of hand shape variation from depth images », in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, vol. 07-12-June, p. 2540-2548.*
(Messaci et al. 2015)	*A. Messaci, N. Zenati, A. Bellarbi, et M. Belhocine, « 3D Interaction techniques using gestures recognition in virtual environment », 2015 4th Int. Conf. Electr. Eng., no ii, p. 2-6, déc. 2015.*
(Oberweger et al. 2015)	*M. Oberweger, P. Wohlhart, et V. Lepetit, « Hands Deep in Deep Learning for Hand Pose Estimation », Proc. 20th Comput. Vis. Winter Work., p. 21-30, 2015.*
(Supancic et al. 2015)	*J. S. Supancic III, G. Rogez, Y. Yang, J. Shotton, et D. Ramanan, « Depth-based hand pose estimation: methods, data, and challenges », arXiv1504.06378, p. 1868-1876, 2015.*
(Marchand et al. 2016)	*E. Marchand, H. Uchiyama, F. Spindler, « Pose estimation for augmented reality : a hands-on survey Pose estimation for augmented reality : a hands-on survey », IEEE Trans. Vis. Comput. Graph., 2016.*
(Taylor et al. 2016)	*J. Taylor et al., « Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences », ACM Trans. Graph., vol. 35, no 4, p. 1-12, juill. 2016.*
(Wu et al. 2016)	*D. Wu et al., « Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition », IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no 8, p. 1583-1597, 2016*
Cheng et al. 2016	*H. Cheng, L. Yang, et Z. Liu, « Survey on 3D Hand Gesture Recognition », IEEE Trans. Circuits Syst. Video Technol., vol. 26, no 9, p. 1659-1673, sept. 2016.*
(KOLKUR et al. 2017)	*KOLKUR, Seema, KALBANDE, D., SHIMPI, P., et al. Human skin detection using RGB, HSV and YCbCr color models. arXiv preprint arXiv:1708.02694, 2017.*
(BELLARBI 2017)	*BELLARBI, Abdelkader. Vers l'immersion mobile en réalité augmentée: une approche basée sur le suivi robuste de cibles naturelles et sur l'interaction 3D. 2017. Thèse de doctorat.*
(Shamayleh et al. 2018)	*Al-Shamayleh, A.S., Ahmad, R., Abushariah, M.A.M., Alam, K.A., Jomhari, N.: A systematic literature review on vision based gesture recognition techniques. Multimed. Tools Appl. 77(21), 28121–28184 (2018)*
(Vuforia 2018)	*Vuforia Developer Library | Portal Pengembang. (2018) Vuforia. [Online]. https://library.vuforia.com/*
(Microsoft 2019)	*Microsoft Visual Studio 2019." Retrieved 05/05/2015, from https://www.microsoft.com/france/visual-studio/evenements/visual-studio2019.aspx.*
(SIGUT et al. 2020)	*SIGUT, Jose, CASTRO, Miguel, ARNAY, Rafael, et al. OpenCV Basics: A Mobile Application to Support the Teaching of Computer Vision Concepts. IEEE Transactions on Education, 2020.*
(ULFAH et al. 2020)	*ULFAH, Syita, RAMDANIA, Diena Rauda, FATONI, Uwes, et al. Augmented reality using Natural Feature Tracking (NFT) method for learning media of makharijul huruf. In : IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2020. p. 012019.*
(asus 2020)	*https://www.asus.com/3D-Sensor/Xtion_PRO/*
(Authorstream 2020)	*Virtual reality. [Online]. http://www.authorstream.com/Presentation/amritaseema518586-vitual-reality/*

(cc.gatech 2020) *HCI issues in Augmented Reality. Google website. [Online]. http://www.cc.gatech.edu/~blair/7001-f02/ar.html.*

(Ilm 2020) *https://www.ilm.com/*

(magicleap 2020) *https://www.magicleap.com/en-us*

(Marxentlabs 2020) *https://www.marxentlabs.com/author/soniamarxentlabs-com/*

(Microsoft 2020) *Microsoft HoloLens, http://www.microsoft.com/microsoft-hololens/en-us (retrieved on 14.3.2015), Microsoft Corporation (Redmont, WA, U.S.).*

(Pokemongolive 2020) *https://pokemongolive.com/en/*

(Wikipedia 2020a) *Virtual reality. [Online]. https://en.wikipedia.org/wiki/Virtual_reality#History*

(Wikipedia 2020b) *Augmented reality. Wikipedia. [Online]. http://en.wikipedia.org/wiki/Augmented_reality#History.*

(Wikipedia 2020c) *https://en.wikipedia.org/wiki/Kinect*