Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics

Human eye gaze following for interaction with a robot in virtual reality

Bachelor Thesis

2024

Ľuboš Hellesch

Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics

Human eye gaze following for interaction with a robot in virtual reality

Bachelor Thesis

Study Programme:	Applied Computer Science
Field of Study:	Computer Science
Department:	Department of Computer Science
Supervisor:	prof. Ing. Igor Farkaš, Dr.

Bratislava, 2024

Ľuboš Hellesch

33151370

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Ľuboš Hellesch
Študijný program:	aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor:	informatika
Typ záverečnej práce:	bakalárska
Jazyk záverečnej práce:	anglický
Sekundárny jazyk:	slovenský

Názov: Human eye gaze following for interaction with a robot in virtual reality
Využitie sledovania pohľadu človeka pri interakcii s robotom vo virtuálnej realite

Anotácia: Interakcia medzi človekom a robotom (HRI) vo virtuálnej realite ponúka množstvo výhod v porovnaní s fyzickou interakciou, no jej potenciál ešte nie je úplne preskúmaný. Sledovanie pohľadu očí pomocou merača zabudovaného vo VR poskytuje pridanú hodnotu pre používateľa, ale aj pre robota v prípade HRI.

Cieľ: 1. Pripravte prostredie Unity, VR s meračom pohľadu očí a simulovaného robota NICO.
2. Vytvorte a implementujte modul umožňujúci robotovi sledovať pohľad ľudského oka pomocou signálov z merača pohľadu očí.
3. Analyzujte softvérové a hardvérové obmedzenia navrhovaného prístupu.

Literatúra: Adhanom I.B., MacNeilage P., and Folmer E. (2023). Eye tracking in virtual reality: a broad review of applications and challenges. Virtual Reality, pp. 1–24.
Clay C., König P., König S. (2019). Eye tracking in virtual reality. Journal of Eye Movement Research. https://doi.org/10.16910/jemr.12.1.3
Sipatchin A., Wahl S., Rifai K. (2021). Eye-tracking for clinical ophthalmology with virtual reality (VR): A case study of the HTC Vive pro eye's usability. Healthcare. https://doi.org/10.3390/healthcare9020180

Vedúci:	prof. Ing. Igor Farkaš, Dr.
Katedra:	FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry:	doc. RNDr. Tatiana Jajcayová, PhD.
Dátum zadania:	19.10.2023

Dátum schválenia: 01.11.2023

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

študent

vedúci práce

33151370

Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Ľuboš Hellesch
Study programme:	Applied Computer Science (Single degree study, bachelor I. deg., full time form)
Field of Study:	Computer Science
Type of Thesis:	Bachelor´s thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Human eye gaze following for interaction with a robot in virtual reality

Annotation: Human-robot interaction (HRI) in virtual reality offers a number of advantages compared to physical interaction, yet its potential has not been fully explored. Measuring human eye gaze embeddded in a VR provides an added value both for the user, but also for the robot in case of HRI.

Aim:
1. Prepare the Unity environment, VR with eye tracker and the simulated NICO robot for HRI.
2. Create and implement a module enabling NICO to follow human eye gaze using signals from eye-tracker.
3. Assess the software and hardware limitations of the proposed approach.

Literature:
Adhanom I.B., MacNeilage P., and Folmer E. (2023). Eye tracking in virtual reality: a broad review of applications and challenges. Virtual Reality, pp. 1–24.
Clay C., König P., König S. (2019). Eye tracking in virtual reality. Journal of Eye Movement Research. https://doi.org/10.16910/jemr.12.1.3
Sipatchin A., Wahl S., Rifai K. (2021). Eye-tracking for clinical ophthalmology with virtual reality (VR): A case study of the HTC Vive pro eye's usability. Healthcare. https://doi.org/10.3390/healthcare9020180

Supervisor:	prof. Ing. Igor Farkaš, Dr.
Department:	FMFI.KAI - Department of Applied Informatics
Head of department:	doc. RNDr. Tatiana Jajcayová, PhD.
Assigned:	19.10.2023
Approved:	01.11.2023

doc. RNDr. Damas Gruska, PhD.
Guarantor of Study Programme

.... Student

.... Supervisor

I hereby declare that I have independently completed the entire bachelor's thesis on the topic "Human eye gaze following for interaction with a robot in virtual reality" including all its attachments and illustrations, using the literature listed in the attached bibliography and artificial intelligence tools. I declare that I have used the artificial intelligence tools in accordance with the relevant legal regulations, academic rights and freedoms, ethical and moral principles, while maintaining academic integrity, and that their use is appropriately indicated in the thesis.

Acknowledgments: I would like to express my gratitude to my supervisor, prof. Ing. Igor Farkaš, Dr. for his invaluable guidance, support and encouragement throughout the development of my thesis.

Abstrakt

Táto štúdia sa zaoberá využitím sledovania ľudského pohľadu vo virtuálnej realite (VR) na zlepšenie interakcie medzi človekom a robotom (HRI). Našim cieľom je umožniť virtuálnemu robotovi NICO vnímať a reagovať na ľudský pohľad v prostredí VR pomocou technológie sledovania očí, čím chceme vytvoriť intuitívnejšie a pútavejšie interakcie. Aby sme dosiahli naše ciele, zaviedli sme systém, ktorý kombinuje sledovanie očí s platformou virtuálnej reality. Účelom tejto konfigurácie bolo vytvoriť interaktívne prostredie v Unity, v ktorom by robot mohol reagovať na ľudské vizuálne podnety, čo umožňuje presnú kontrolu prostredia a vedie k intuitívnejšiemu procesu interakcie. Vykonali sme malý pilotný test dostupných technológií zameraný na imerziu a spoľahlivosť počas sledovania ruky a ovládača. Výsledky poskytujú prehľad o výhodách a ťažkostiach používania sledovania pohľadu vo VR pre interakciu medzi človekom a robotom. Tiež objasňujú proces a zdroje potrebné na vytvorenie takéhoto systému, pričom zdôrazňujú potenciálne obmedzenia v súvislosti s kalibráciou hardvéru, softvérovou latenciou a presnosťou sledovania pohľadu.

Kľúčové slová: meranie pohľadu očí, sledovanie pohľadu očí, virtuálna realita, Unity

Abstract

This study explores the use of eye tracking in virtual reality to enhance human–robot interaction. Our goal is to enable the virtual robot NICO to perceive and respond to human gaze in a VR environment using eye tracking technology, aiming to create more intuitive and engaging interactions. In order to achieve our goals, we put in place a system that combines eye tracking with a virtual reality platform. The purpose of this configuration was to create an interactive environment in Unity, where the robot could react to human visual cues, to enable precise control of the environment resulting in a more intuitive interaction process. We performed a small pilot test of the available technologies focused on immersion and reliability during hand and controller tracking. The results provide insight into the benefits and difficulties of utilizing eye gazing in VR for human–robot interaction. They also clarify process and resources needed to create such a system while revealing potential limitations in terms of hardware calibration, software latency and gaze tracking accuracy.

Keywords: eye tracking, eye gaze following, virtual reality, Unity

Contents

List of Figures

List of Tables

List of Abbreviations

VR Virtual Reality
HRI Human–Robot Interaction
HMD Head–Mounted Display
NICO Neuro–Inspired Companion
HRC Human–Robot Collaboration
AR Augmented Reality
AV Augmented Virtuality
VOG Video–Octulography
SDK Software Development Kit

Introduction

The field of robotics has evolved over several millennia, without reference to the word robot until the early 20th century. When the word was derived from the Czech word robota which means servitude or forced labour, the origins of mechanical devices can be traced back to ancient Greece in 270 B.C. These early inventions paved the way for modern robotics as we know it today (Kurfess et al., 2005). The passion driving people to make their jobs easier was always there. It is one of the main reasons behind humankind's fast-paced technological advancements continuing to this day. The first robots in general, were machines or devices that operated automatically or by remote control. Based on the premise of making the tasks easier, they were designed to require as little human intervention as possible. People have wanted to use such equipment since simple devices were developed (Kurfess et al., 2005). However, this is only part of the spectrum of things that the word robot describes. There is also, nowadays not so much, science fiction version, that is usually more human-like with the ability to precisely replicate human behaviour and abilities.

With the advances in the field of artificial intelligence. We can already see its implementation into many types of robots, even some of those science fiction humanoids or androids. Many of them now operate autonomously, without human intervention as before, simply completing tasks more efficiently. There is an incentive to create robots, that can to a certain level replicate human movements, thinking and emotion. This could be very useful in lots of ways and we can see it as the fast-approaching future of many different industries. That incorporates the need to interact more intuitively with these types of robots.

Human-robot interaction strives to enable easy, intuitive interactions between people and robots. Such interactions require natural communication. Although verbal communication tends to be primary in human-human interactions, nonverbal behaviours, such as eye gaze and gestures, can convey mental state, augment verbal communication and reinforce what is being said as shown earlier (Breazeal, 2004). The correct integration of these principles can be regarded as the most important part of integrating robotics into the day-to-day lives of people.

Virtual reality creates the perfect environment for implementing, testing and studying these principles. It has been the subject of studies since the beginning of the 20th century. Today, we can already observe the results of many of them in practice. There is no doubt about the importance of VR currently or in the future. The conveniences offered by VR are now accessible to the average user, as this technology has become generally available. VR enables users to immerse themselves in diverse virtual environments with the use of head-mounted display (HMD) devices. Same as with HRI one of the key components of a realistic VR experience is precise and reliable user interaction.

Nonverbal signals have proven to be an important tool for improving interaction and other aspects of VR or HRI applications. Eye gaze is a particularly important nonverbal signal, compared with pointing, body posture and other behaviours, because evidence from psychology suggests that eyes are a cognitively special stimulus with unique hard-wired pathways in the brain dedicated to their interpretation (Admoni and Scassellati, 2017).

The objective of this thesis is to explore the potential of HRI within VR environments, with a specific focus on measuring and utilizing human eye gaze. We aim to highlight the advantages of VR-based HRI over physical interactions by emphasizing the unique benefits offered by VR. The thesis describes the setup and preparation required to support VR and eye-tracking functionalities. We implement a module that enables the NICO robot to follow the human eye gaze by using signals from the eye tracker. The technical aspects and methodologies used to create this module are thoroughly explained. Finally, we make an assessment of the software and hardware limitations encountered during the implementation and identify potential challenges and suggest possible solutions or areas for further research.

Chapter 1

Human–Robot Interaction

This chapter explores the field of HRI, briefly exploring its historical evolution, examining key implementations from early experiments to the first applications and describing state-of-the-art development.

1.1 Introduction

HRI is a field of study that focuses on the design, understanding and evaluation of robotic systems as they interact with humans. This interdisciplinary field connects robotics, artificial intelligence, engineering, psychology and cognitive science to explore and optimize interactions between humans and robots. While human-automation interaction, such as in the context of piloting aircraft, has been a longstanding and active area of study, genuine HRI did not receive significant research attention until the 21st century. This disparity highlights the historical focus on automation systems and the comparatively recent focus on developing and understanding the complexities of interactions between humans and robots. As technology advanced, the need for comprehensive research in HRI has become increasingly apparent, prompting a surge of interest and investigation in this domain (Sheridan, 2016).

1.2 History

The historical development of HRI goes from way back in the past to the current time, with evolving technology and expanding applications, as robots have transitioned from the pages of science fiction into physical entities that work together with humans in everyday tasks. This introduction to the history of HRI traces the progression from early robotic concepts to the sophisticated interactions we see today, highlighting how shifts in technology, society and academic focus have shaped the field.

HRI is closely tied to robotics and robots which, as we mentioned earlier, have been the centre of development and research since ancient times. Even the historical devices from ancient Greece and those after them were designed for human intervention in some ways. The concept of a robot has been around for a very long time. At first, some of these machines were referred to as automata. However, the designation for these devices was not final, until 1921 when Karel Čapek, a Czech playwright, came up with an intelligent, artificially created person, which he called a robot. Then it was not until the 1940s that the modern-day robot was created with the arrival of computers. After that, the term robotics, the study and use of robots, came about in 1941 and was first adopted by Isaac Asimov, a scientist and writer as shown in (Hagis, 2003). Then the late 20th century marked the fast spread of robotics across many industries and rapid progress.

To provide a clearer brief insight into the complicated history of HRI we divided it into following parts.

First machines

The concept of robotics, in its earliest forms, can be traced back to mechanical devices created in ancient civilizations. In the Middle Ages we know about water-driven devices, including automatic musical instruments and the most sophisticated clock of that time (Romdhane and Zeghloul, 2009). Towards the end of the 18th century, the industrial, efficiency and cost-saving potential of automatic machines began to be explored.

Industrial revolution and robots

During this period interest in robots was rising. In Europe we saw intricate machines able to write, draw, or play music. These robots were not only public curiosities but also further demonstrations of the mechanical possibilities before first electronic systems (Russell, 2017). In the second industrial revolution, electricity transformed manufacturing, making mass production possible with assembly lines. This period marked the start of what we now recognize as traditional industrial robots. The automation of various tasks not only increased efficiency and productivity but has also paved the way for further innovations in robotics and automation. By taking over difficult and repetitive tasks, these robots began to transform the industrial landscape and demonstrate the impact of integrating robotic systems into manufacturing and other sectors.

Modern robotics and World Wars

In the early 20th century, in the United Kingdom, several remote-controlled, full-sized humanoid robots were developed. These robots constructed with multi-part metal bodies and limbs were capable of sophisticated movements. They were primarily designed and built for public display and entertainment. This innovative design of these robots highlighted the potential of robotics in public entertainment and showcased early advancements in remote-controlled humanoid robots (Russell, 2017).

Later, following the World War II, the world experienced a strong industrial push, reviving the economy. Rapid advancement in technology drove this industrial wave and pushed forward the creation of new devices like servos, digital logic and solid-state electronics. The merger of this technology and the world of science fiction came in the form of the ingenuity of George Devol. His team of engineers created the first industrial robot. It was named the Unimate (Fig.1.1) (Kurfess et al., 2005).

Figure 1.1: A picture of the Unimate first prototype (Dafarra, 2020).

Expansion and diversification

The latter part of the 20th century saw robotics spread across industries, with rapid advancements in technology and increased commercial production of robotic systems. This era introduced the standardization of robotic systems as articulated arms with varying degrees of freedom. They were used for a range of applications not only in industry but also in laboratories.

Intelligent and autonomous robots

With the advances in computing power and AI, robots can now perform complex tasks requiring decision-making and real-time responses. It's linked to major advances in robotics and HRI. These advancements have enhanced robots abilities to perform complex tasks and interact with humans in more sophisticated ways. This intensified the focus on creating robots capable of meaningful human interactions. The aim is to develop robots that can seamlessly integrate into human environments, performing tasks that require advanced social and cognitive skills. A notable example is the Neuro–Inspired Companion (NICO) project (Fig.1.2), which strives to build an affordable platform that simulate human emotions and behaviors, enhancing natural and effective interaction with people ((Kerzel et al., 2017)).

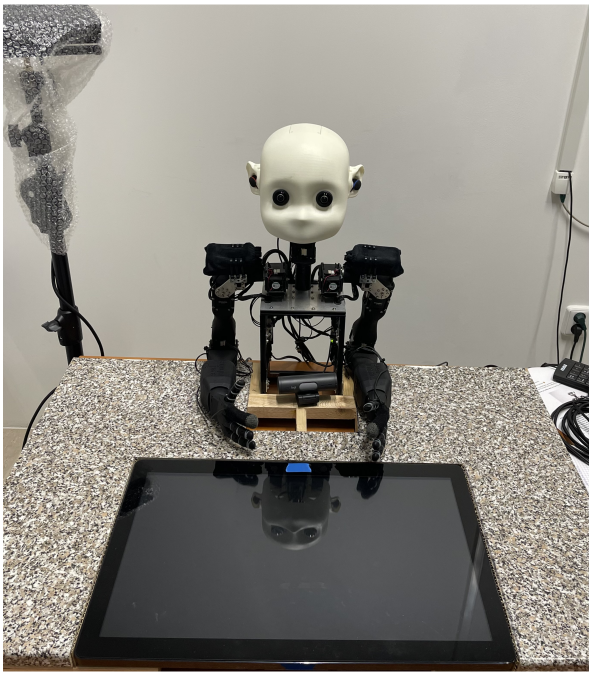


Figure 1.2: A picture of robot NICO at our faculty.

Future directions

The future of robotics is set on enhancing the interaction between humans and machines, developing machines that can operate in complex social and natural environments. The research is focusing on human–robot collaboration in workplaces and AI and ensuring that robots operate within socially accepted norms. The integration of robotics with other emerging technologies like the Internet of Things and big data companies is expected to further expand the capabilities and applications of robotic systems.

1.3 Human–Robot Interaction

Examples and applications are seen through the history and expected future. We can split the various models of interaction between humans and robots into these: teleoperation, supervisory control, collaborative control, autonomous interaction and sociable interaction.

Each model reflects different aspects and goals of HRI, from enhancing human capabilities to fostering more natural and efficient human–robot collaboration (HRC).

1.3.1 Teleoperation

Teleoperation and Telepresence together are the main part of research into Telerobotics.

Telerobotics is a branch of robotics that focuses on the remote control of semi-autonomous robotic systems, primarily through wireless communication. The main challenge in telerobotics is to establish an optimal interface between the human operator and the robot, ensuring that the user receives highly accurate and reliable data from the robot's sensors. Furthermore, it is essential to provide the user with an easy-to-use and intuitive control interface, designed to maximize the efficiency and effectiveness of the desired operations.

Teleoperation is the designation for controlling objects or robots remotely. However in general it is used to describe objects that need to be controlled constantly and directly by an operator.

Telepresence refers to an event where an individual feels as though they are present in a distant location or time, different from their actual physical position. In telerobotics, this is the combination of technologies that enable a person to experience the environment of the remotely controlled robot. Incorporating the interaction with an environment that can be real, virtual or a combination of both (Čorňák, 2020).

1.3.2 Supervisory control

Supervisory control involves a high-level monitoring and management strategy where a human operator periodically interacts with remote automated systems (Čorňák, 2020). These systems function autonomously between these supervisory interventions, maintaining operations independently while the human provides strategic guidance and oversight as needed. This model of interaction is mainly used to operate robots in situations where full automation is not possible due to unpredictability, complexity of the environment or legal reasons mainly associated with safety.

1.3.3 Collaborative control

Collaboration or HRC is a field engaged in the research of cooperation between humans and robots. As it is a type of interaction, HRC is a sub-field of more general HRI. The HRI is more focused on the more universal connection between humans and robots which may or may not result in profit for either side. While HRC is based on the interaction of humans and robots which work together to achieve a common goal. In HRC humans and robots create teams, with predefined responsibilities on each side to ensure the work is effective and safe. Each subject has to be familiar with intentions of each other and greatly familiar with the environment they collaborate in (Čorňák, 2020). The role of humans and robots in collaborative tasks depends on both the nature of the task and the level of interaction required. According to the International Federation of Robotics, there are five distinct levels of collaborative HRI (Fig.1.3) (Hroboň, 2021):

- *Work cell*: The human and robot work on a common task, but their workspaces are separated by a protective robot cage and they perform tasks independently.
- *Coexistence*: The human and robot work on a common task side by side without physical barriers like cages, but do not share the same workspace.
- *Sequential* collaboration: The human and robot share the same workspace, but they perform tasks sequentially.
- *Cooperation*: The human and robot share the same space and work on the task simultaneously but independently.
- *Responsive* collaboration: The robot responds to the human's movements in real time.

Collaboration robots currently belong to a rapidly growing sector, both commercially and scientifically. They are part of the emerging industrial revolution. These robots can also branch off into different sectors like healthcare, educational system or services (Hroboň, 2021).

1.3.4 Autonomous interaction

Autonomous robots operate independently, making decisions and performing tasks without human intervention. These robots are designed to adapt to dynamic environments and handle unexpected situations. The model emphasizes the development of sophisticated AI and machine learning algorithms to enable full autonomy. Nowadays we can see these types of robots as automated vehicles in which a human is a passenger, as well as automated highway and rail vehicles and commercial aircraft.

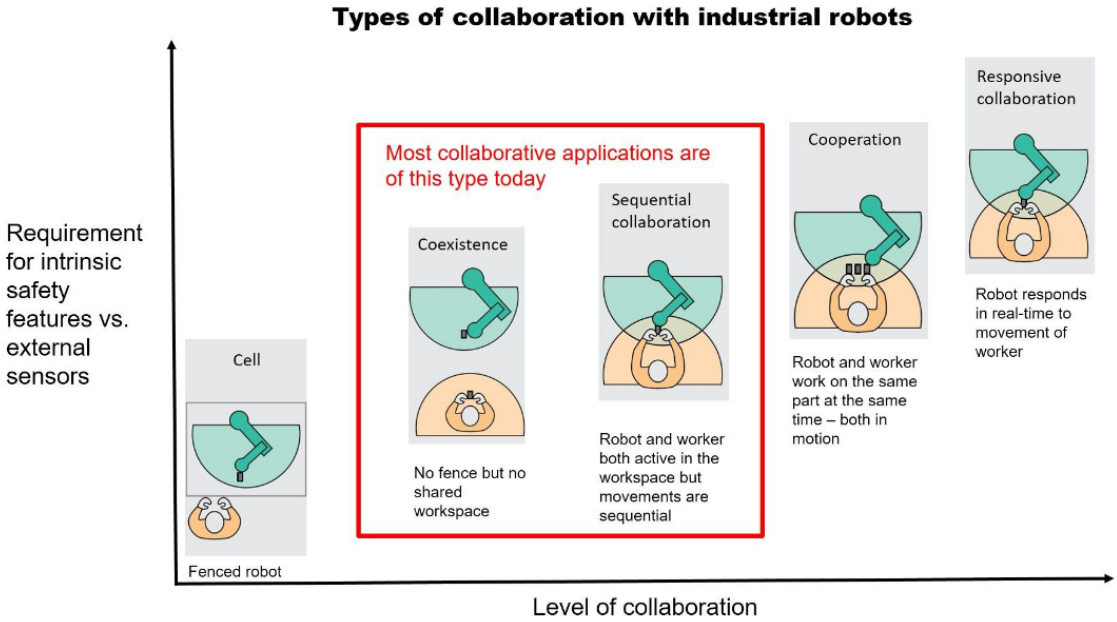


Figure 1.3: Levels of collaboration with an industrial robot (Hroboň, 2021).

The DARPA Grand Challenge contests demonstrated the proven AI-guided autonomous vehicles. For instance, Google's self-driving car has successfully managed to navigate California freeways as shown earlier (Sheridan, 2016). Furthermore, Tesla, a more recent leader in autonomous driving technology, has introduced vehicles capable of operating autonomously under various conditions. These cars are already available to the general public, demonstrating significant advancements in making autonomous driving technology widely accessible. Tesla's approach not only showcases the practical application of autonomous systems but also emphasizes the growing integration of technologies in everyday transportation. However, they are always expecting human drivers to remain alert and ready to take over in case of automation failure.

Other automobile manufacturers chose the path of developing technologies to assist drivers, such as radar-augmented cruise control, run-off-the-road alarms and vehicle-to-vehicle communication. Complex traffic situations remain difficult for AI to understand, as many accidents are avoided through social interactions between drivers. Research is needed to understand the social aspects of driving and the extent to which cars can be safely automated (Sheridan, 2016).

1.3.5 Sociable interaction

This model focuses on robots that engage with humans in social contexts. These robots are designed to understand and respond to social cues, providing interactions that are more natural and intuitive. Applications include companion robots, educational tools and interactive entertainment.

The biggest question in this kind of interaction is how to properly interface untrained humans with these technologies in a way that is intuitive, efficient and enjoyable to use. Crucial in the development can be Human-Computer Interaction principles and the enhancement of sociable robots. Sociable robots are designed to engage with humans on a social level, utilizing expressive features and behaviours that facilitate natural and intuitive interactions. In the next parts, we talk about the key points of sociable robotics.

Understanding and emulating social cues

The first robot to use expressive features to convey emotions and respond appropriately to human facial expressions and voice tones was from the Massachusetts Institute of Technology named Kismet (Fig. 1.4). This ability to understand and replicate social cues is essential for creating natural HRI. However, we can see that not only the ability to perceive and respond to emotions correctly but the design of robots is also very important (Breazeal, 2004). Key hardware components like eye trackers, microphones and cameras, play a big role in the process of understating emotions. For a robot to respond accurately to various social cues from humans, it must be provided with sufficient information about those cues.

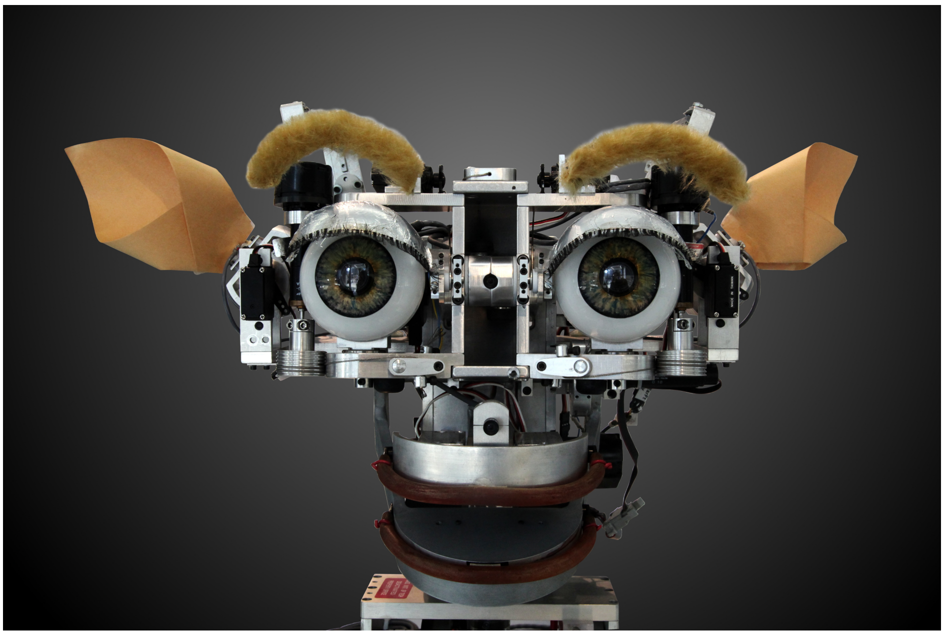


Figure 1.4: A picture of robot Kismet from MIT (Rama, 2019).

Learning from human feedback

Sociable robots are programmed to adapt their behaviours based on human feedback, refining their social skills through interactions. This adaptive learning process makes their responses more contextually appropriate and improves overall interaction quality.

Applications across domains

Sociable robots can be employed in various fields, including education, healthcare and entertainment. They assist in teaching through interactive play, support elderly patients with daily tasks and provide companionship to reduce loneliness.

Design challenges

Developing sociable robots involves addressing challenges such as creating lifelike expressions, ensuring intuitive user interactions and maintaining user engagement. Properly integrating all advanced software and hardware is integral to overcoming these challenges and enhancing the robot's social capabilities. With proper design features to make the interaction with robots enjoyable.

Future research and development

Continuous development is critical for improving the social abilities of robots. This includes studying human social behaviour, advancing robot learning algorithms and developing better human–robot communication interfaces.

An important tool for better and faster development of any robots based on HRI is VR. It offers several significant advantages for the development. Firstly, VR can create highly immersive and controlled environments, enabling researchers to study human social behaviour in various scenarios with precision. This immersive setting allows for the detailed tracking and analysis of subtle human behaviours, such as eye movement, body language or facial expressions, which are essential for understanding social interactions. Additionally, VR facilitates the training of robots in virtual environments, eliminating the risks associated with real-world testing. This safe and controlled platform enables humans to interact with robots and provide real-time feedback on their performance. Moreover, VR environments can be dynamically adjusted, allowing researchers to test how robots adapt to new situations and challenges effectively.

Chapter 2

Virtual reality

In this chapter, we dive into the field of VR. Briefly talking about history, its purpose and utilization. Then we provide insight into the technology that supports all that.

2.1 Introduction

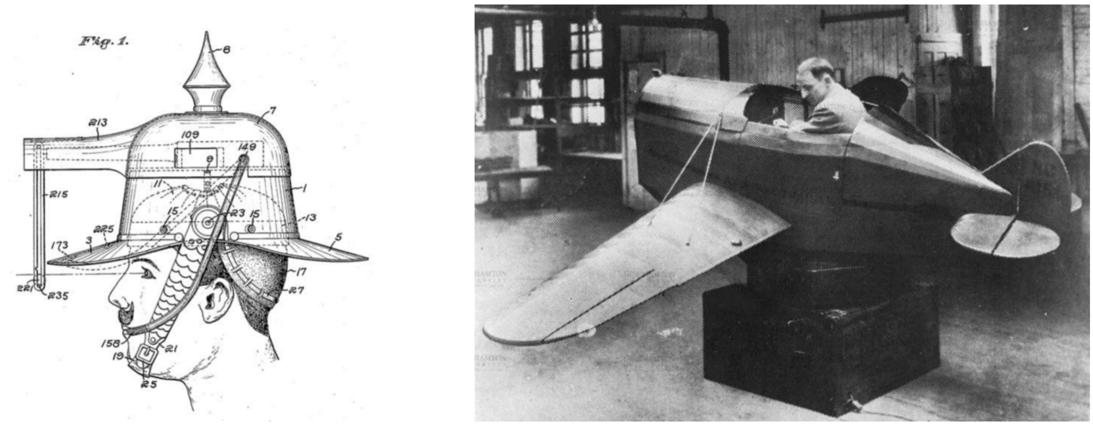
VR is defined as a computer-generated digital environment that can be experienced and interacted with as if that environment was real. It refers to the use of computer technology to create a simulated environment that can be interacted with in a seemingly real or physical way by a person using special electronic equipment, like a headset with a screen inside with controllers, gloves fitted with sensors or any devices that can provide deeper immersion within any virtual environment. Unlike traditional user interfaces, VR places the user inside an experience, allowing them to interact with three-dimensional space (3D) worlds. Ivan Sutherland, the creator of one of the world's first VR systems, held the opinion that the perfect VR system enables its users to physically walk around the objects and feel them as real. The computer would have control over the existence of matter (Jerald, 2015).

2.2 History

Predecessors of what we call VR today, go as far back as the 19th century. Inventions like stereoscope which in its time caused the 3D craze, with smaller cardboard versions available to the general public. Stereoscopes of that time were very similar to Google Cardboard VR and phone-based VR systems alike. But instead of phone those used pictures.

The development in the 20th century goes beyond just showing pictures. Some important interaction concepts were created, like head-worn gun-pointing devices in 1916 (Fig.2.1(a)), or the first simple mechanical flight simulator from 1928 created by

Edwin Link (Fig.2.1(b)).



(a) (b)

Figure 2.1: a) The head-worn pointing device, (b) the first flight simulator (Jerald, 2015).

Together with the technological inventions of this era, the first science fiction stories emerged, exploring questions about the nature of reality. For instance, in 1935, a book described a future very similar to what we now try to achieve, including an HMD that replaces reality with an artificial one in a story centered solely around the individual user. This early speculation laid the groundwork for the concepts and technologies that continue to shape our understanding of virtual and augmented realities today (Jerald, 2015).

The first patents for HMD, as described in science fiction, came in 1950 by Morton Heiling (Fig. 2.2). The HMD is described is using lenses capable of one hundred forty-degree horizontal and vertical field of view and stereo speakers. His other, similarly interesting, patent from the same time was Sensorama (Fig. 2.3), a world-fixed display. It was meant to make watching movies more immersive with things like seat vibrations, wind, stereo sound and scent (Jerald, 2015).

Following these initial developments, the first functional HMD was introduced in 1960. This system was designed to translate the user's head movements into camera movements in a separate room. This innovation marked a significant milestone, as it represented the first demonstration of the potential of telepresence technology (Sec. 1.3.1). After that, numerous devices were developed, ranging from IBM's first glove input device to various HMDs created for military applications. These innovations built on the foundational work of early telepresence systems, expanded the possibilities for immersive interaction and remote control. Each new device contributed to the development of virtual reality and teleoperation, highlighting the rapid advancements in technology and its growing applications across different sectors. Subsequent development by engineers at NASA brought the first commercially available HMD called

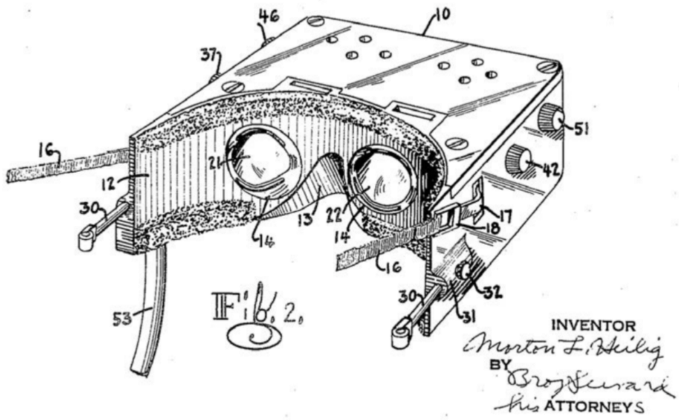


Figure 2.2: The drawing of Stereoscopic Television Apparatus on the patent by Helling (Jerald, 2015).

Virtual Visual Environment Display. With an additional development of a system, that provided 3D sound and a relatively affordable production price, they started the VR industry. Despite the initial excitement and numerous predictions about the immense potential of the VR industry, interest in VR significantly declined by the end of the century. This decline was largely due to the technological limitations of the time, which were unable to support and fulfil the great visions and expectations of the public. The difference between the imagined possibilities and the actual capabilities of VR technology led to a reduction in enthusiasm and investment in the field (Jerald, 2015).

VR came alive in the 21th century around Kickstarter by Oculus VR and their product Oculus Rift HMD (Fig.2.4). With great interest from the public and big companies, the new era of VR was born (Jerald, 2015).

2.3 Types of reality

There are various forms that can be divided into a range from real environments to virtual ones. Intermediate forms, falling between virtual and augmented reality, are known as mixed reality, which includes both augmented reality (AR) and augmented virtuality (AV). The real environment represents the world we live in. While replicating real-world experiences isn't always the goal of VR, understanding how we perceive and interact with the real world is crucial for developing relevant functionalities for VR experiences. The concentration on these functions depends on the specific goals of the application. So we need to know more about how we perceive real environments to be able to create more immersive virtual environments.

AR only enhances the real world by overlaying computer-generated objects, ideally

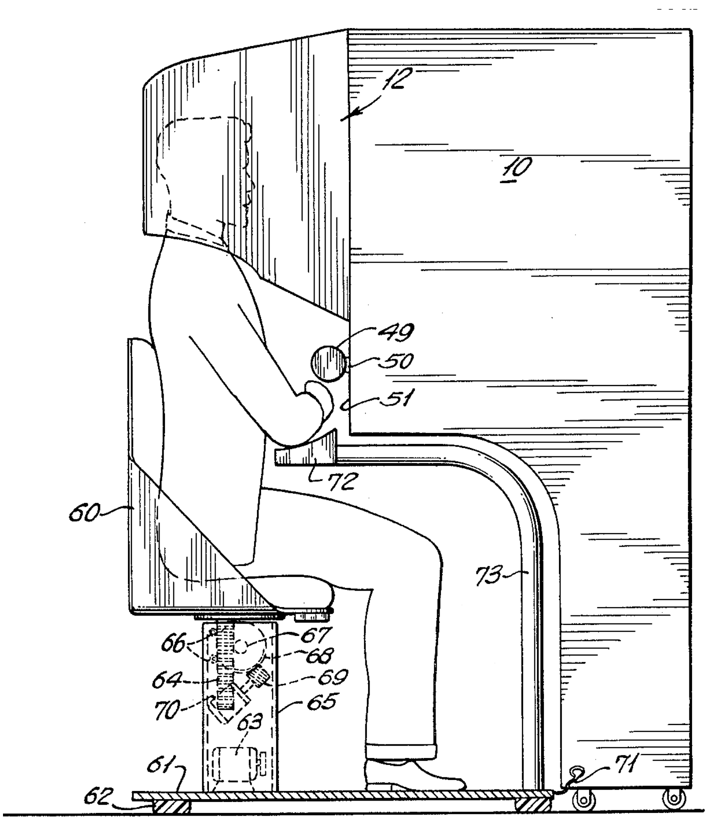


Figure 2.3: The drawing of world-fixed display Sensorama by Helling (Jerald, 2015).

making it indistinguishable from reality to the human mind. AV involves integrating real-world content into VR, such as immersive films, recordings can range from single viewpoints to complex light fields or geometries, allowing users to navigate the environment freely.

In contrast, true virtual environments are fully artificial, created without any real-world content. The aim of these environments is to fully immerse the user, making them feel present in an entirely different world while minimizing any negative effects.

2.4 Bringing VR closer to reality

To achieve the best possible VR experience, many factors are required, like proper audio, haptics, and most interesting for us, visual quality. It is estimated that to enable such display systems, we would need to deliver more than 100 Gb/s of data to the HMD (Bastani et al., 2017). Achieving such transmission speeds is not only challenging, but as various studies addressing this issue have found, unnecessary. Since human vision has the highest acuity only in the foveal area, which covers only four percent of typical VR displays, we can reduce the sharpness in the ninety-six percent of the image without affecting the perceived quality as shown earlier (Weier et al., 2017). If we use high-quality data obtained from eye tracking, we can drastically reduce

Figure 2.4: VR headsed by Oculus VR (Evan-Amos, 2017).

hardware requirements or increase quality or refresh rate. According to Sipatchin et al. (2021), we can increase performance by up to seventy percent when applying the above-described procedure. Such a reduction in hardware load is used, for example, when displaying 360-degree video, which has high demands on image quality and allows us to view the scene from a central perspective, providing a VR experience that approaches reality. This could provide better immersion and therefore create a better virtual world.

2.4.1 Immersion and presence

Immersion describes the extent to which a user feels present in a digitally created world. High levels of immersion can create feelings for the user as if they have physically entered and are part of the virtual environment. However, immersion is only a part of the VR experience as humans have to perceive and intercept presented environments. How a user experiences the immersion is called presence.

In the context of VR, presence is described as the feeling of being in some place even when being physically located in a different place. This can be described as one of the main goals of VR. Immersive technologies that engage the user's senses and create realistic interactions are a good basis for the feeling of presence, but it is not always so. However, presence is limited by immersion as shown earlier (Jerald, 2015).

2.5 Eye movement based interaction in VR

Measuring gaze direction in real time can be used as a useful way to interact with various systems. In VR gaze-controlled interface systems can have many applications not only due to more efficient control but also for accessibility for physically handicapped people. Recently, the American company Apple made a big step towards integrating these principles into people's daily lives. At the beginning of June 2023 they introduced an HMD (Fig.2.5) that is controlled exclusively by gaze, hand gestures and voice.

Figure 2.5: Apple Vision Pro AR HMD (Kranen, 2024).

2.6 Applications in various fields

In this section, we talk about some of the different applications of VR and eye tracking in VR.

2.6.1 HRI

VR makes the development and training of robots easier and more available by providing a safe and controlled environment for testing and interaction. Researchers can observe and analyze robot behaviour in various scenarios without the risks associated with real-world testing.

2.6.2 Research and development

In research, VR is used to study human behaviour under various conditions, offering a controlled environment where variables can be precisely manipulated. This is particularly useful in psychology, social sciences and any other research field that benefits from precisely controlled environments. Which could be easily created.

2.6.3 Education

Educational institutions and corporations use VR to create interactive learning experiences and simulations that enhance understanding of complex subjects. This includes virtual laboratories, historical recreations and safety training programs (Voštinár et al., 2021). VR simulations provide these environments for various purposes, eye tracking data can be used to measure attention, identify areas of interest or potentially improve the efficiency of work environments.

2.6.4 Medicine

VR provides immersive training environments for medical students and professionals, allowing them to practice procedures and surgeries in a risk-free virtual setting. This enhances their skills and confidence before performing real-life operations. It is also revolutionizing medicine by providing immersive training environments for medical professionals and aiding in patient rehabilitation through therapeutic exercises that improve mobility and strength (Bruno et al., 2022). The eye tracking shows potential in medicine as well. It is already used as an objective metric in the treatment of generalized anxiety disorder and various phobic disorders. It will certainly find its application in ophthalmology as well as shown earlier (Sipatchin et al., 2021).

2.6.5 Entertainment

The gaming industry is one of the most prominent adopters of VR, with numerous VR games and experiences offering players an unprecedented level of immersion. VR is also used in theme parks and virtual tours to provide great entertainment experiences (Voštinár et al., 2021).

Chapter 3

Eye gaze tracking

In this chapter, we will talk about the importance of gaze following in HRI and introduce the basic principles of eye tracking.

3.1 Eye movements

Understanding different types of eye movements is crucial for effective eye tracking. Optimizing eye tracking often requires identifying and differentiating various types of eye movements and utilizing knowledge about visual and cognitive processing during eye movements to achieve the desired outcome (Adhanom et al., 2023).

The basic types are saccades, smooth pursuit, vestibulo-ocular reflex and vergence eye movements.

- Saccades are rapid movements of the eyes that shift the line of sight from one point to another. These movements are essential for efficiently scanning the environment, reading and other tasks that require quick shifts in attention. By allowing the eye to reposition quickly, saccades enable us to sample different parts of the visual field. The crucial work of vision happens during the brief periods of relative retinal stability between successive saccades, when the eye is stationary, allowing for high visual acuity and detailed perception (Kowler, 2011).

- Fixations occur when the eyes are stationary and focused on a single point. Fixation duration, typically between 100 to 300 milliseconds, is a critical measure in eye tracking as it indicates what is being attended to and processed visually (Farnsworth, 2018).

- Smooth pursuit refers to smooth tracking of selected, and typically foveal, targets (Kowler, 2011). Those are slow and involuntary movements that allow the eyes to follow a moving object smoothly. This type of movement is essential for activities like watching moving objects or tracking a target (Farnsworth, 2018).

- Vestibulo-Ocular Reflex stabilizes the gaze during head movements by producing eye movements in the opposite direction of the head movement. It's vital for maintaining stable vision during dynamic activities like walking or driving (Wang and Li, 2024).

- Blinks, while they are often overlooked, blinks can provide valuable data in eye tracking, especially in understanding cognitive load and fatigue (Wang and Li, 2024).

- These are movements where both eyes move in opposite directions to obtain or maintain single binocular vision. Vergence movements coordinate fixation at different depths, which is more accurately depicted in the figure (Fig. 3.1) (Kowler, 2011).

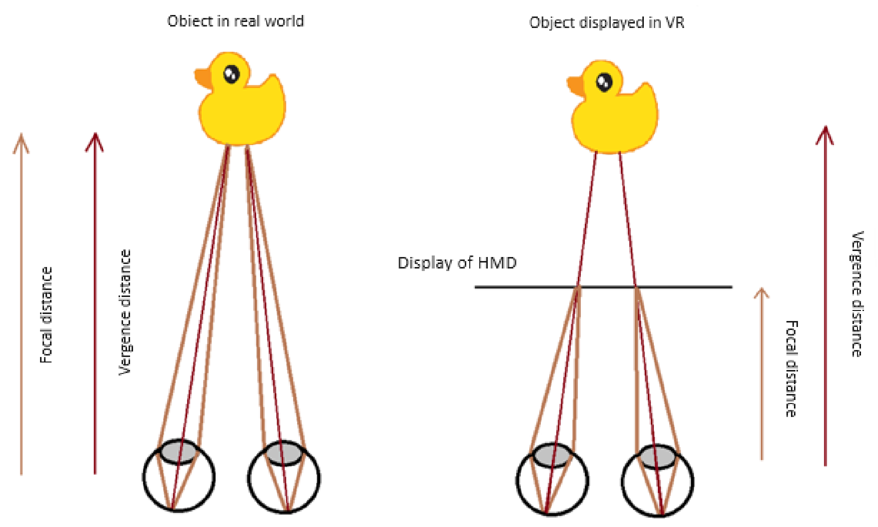


Figure 3.1: The difference between VR and the real world is important for gaze fixation at different depths (Clay et al., 2019).

An eye tracking device must be precise enough to correctly recognize these movements, which is not entirely trivial, as several of these eye movements can occur simultaneously. So the overall eye movement is the result of a superposition of interacting movements, complicating the identification and quantification of different types of eye movements. Additionally, it is necessary to also work with head movement, since gaze shifts larger than thirty degrees are typically achieved by a combination of saccades and head movement (Adhanom et al., 2023). To accurately measure all important factors, we need to utilize advanced methods for processing information.

3.2 Gaze following in HRI

Gaze following is a key part of social eye gaze, a fundamental aspect of human communication. Important factors include the direction and duration of eye movements during social interactions. Social eye gaze includes eye movements that are intentionally expressive, such as gaze aversions, which are meant to show thoughtfulness. Additionally, it includes eye movements serving non-communicative purposes, like orienting a robot's view towards an object of interest, provided these movements occur within an interactive context where they may be perceived by others. These movements play a critical role in non-verbal communication, enabling the conveyance of complex social signs and intentions during interactions (Admoni and Scassellati, 2017).

- Mutual gaze or direct eye contact between the robot and human, enhances engagement and makes the robot appear more social and intelligent. It helps to create a connection and improve interactions.
- Referential gaze which the robot uses gaze to direct human attention to specific objects or locations, aiding in tasks such as collaborative work or instruction.
- Joint attention is shared focus on an object or event, crucial for tasks requiring cooperation and coordination. It is particularly beneficial in educational and therapeutic settings, such as robot-assisted therapies for children with autism.
- Gaze aversion can be used by robots to signal disengagement or to indicate that the robot is processing information, helping to manage turn-taking in conversations.

A significant part of research on social eye gaze has originated from the virtual agents community. Virtual agents allow for precise control over the appearance and timing of gaze behaviours, including subtle movements of the eyelids, eyebrows and eyeballs. These movements are challenging to replicate using physical motors in embodied robots. Virtual agents can more effectively simulate the delicate aspects of the human eye gaze essential for natural and expressive interactions in virtual environments. Research in this area focuses on optimizing these gaze behaviours to improve the effectiveness of robots in social, educational and therapeutic roles. Developing biologically inspired eye movements in robots can increase their acceptance and effectiveness in human environments (Admoni and Scassellati, 2017).

3.3 Eye movements in VR

To obtain all the information necessary for the proper use of eye movement tracking as described in the previous section, we need integrated sensors, which are available in the HTC Vive Pro Eye HMD that we have the opportunity to work with, or separate sensors that are inserted into the devices. The most commonly used method is video-oculography (VOG), which is also used by the device we are using. Devices working based on VOG technology capture images of the eyes using cameras and then process them. All this information can be used, for example, to determine where or at which object the subject using the VR HMD is looking, to mitigate unwanted effects such as nausea and many other applications.

3.3.1 Determining Gaze in VR

When determining where a subject is looking in the real world, it is necessary to find the direction using the eye gaze vector in three dimensions, originating from the eyes in the direction of the gaze and the depth of gaze, which can be calculated from the divergence of the two eyes by calculating the point of intersection. This method relies on almost unattainable accuracy at greater distances, so we need to use another methodology (Clay et al., 2019).

To determine the gaze point in VR, we utilize the fact that we have much more information about the world being observed, as well as the subject observing it. Using a 3D model of the eye, the distance between the eyes, and objects, we can easily calculate the depth of a point in space (Clay et al., 2019). Calculating the 3D gaze vector is relatively simple depending on the environment in which we are working. This can be done either for each eye individually or using the central eye position.

With the advent of VR HMDs equipped with built-in eye tracking sensors, this process has been significantly simplified for appropriately selected applications. Most VR HMDs can automatically determine which object the subject is looking at, although not always with complete accuracy (Richard, 2021).

3.3.2 Quality of Eye Tracking

In eye gaze tracking, the quality of the data obtained is crucial. We assess this quality through various metrics, including spatial precision, spatial accuracy, sampling frequency, and latency. Spatial precision refers to the eye sensor's ability to consistently reproduce measurements over time, while spatial accuracy indicates the deviation between the actual gaze position and the position estimated by the sensor. Sampling frequency measures how often the sensor captures data, and latency tracks the time delay between the occurrence of eye movement and its processing by the sensor.

To achieve optimal results, proper calibration of the sensors is essential. Calibration is the process by which the eye tracking system determines a mapping function that aligns the coordinates reported by the sensors with the actual gaze points in the visual environment (Clay et al., 2019).

Chapter 4

Methodology

In this chapter, we describe the software, hardware and procedures used to achieve our objectives. We detail the specific tools and technologies employed, explaining their roles and functionalities in the overall implementation process.

4.1 Hardware

Other than computers provided by faculty we utilized the devices described below.

4.1.1 HTC VIVE Pro Eye

The HTC VIVE Pro Eye (Fig. 4.1) is a high-end virtual reality headset designed to deliver highly immersive VR experiences. Released in 2019, it boasts high-resolution displays, integrated headphones and advanced motion and eye tracking capabilities. This device is well-supported by developers, providing extensive guides and resources, which is advantageous for our implementation. The comprehensive support ensures that we can effectively utilize the headset features and easily integrate them into our project.

4.1.2 Tobii eye tracking

The Tobii eye-tracker (Fig. 4.2) is integrated into our HMD to monitor the user's eye movements and gaze direction accurately. Tobii is a leading company in the field of eye tracking technology, offering both hardware and software solutions for a variety of applications. They also provide extensive guides and documentation, essential for ensuring the proper functionality and integration of the eye tracking system.

In the context of VR, it enhances the VR experience by allowing applications to respond to where the user is looking. This can be used for interface control, interaction with virtual objects and gathering user behaviour data.

Figure 4.1: HTC Vive Pro Eye.

4.1.3 Leap Motion

To enhance immersion in our project, we utilized hand tracking technology from Ultraleap. Specifically, we utilized the Leap Motion tracker (Fig. 4.3), renowned for its precision and responsiveness in capturing hand movements. Ultraleap is a leader in this domain, providing robust support and comprehensive resources for developers. This extensive support ensures the seamless integration and optimization of the hand tracking system, ensuring that the technology functions effectively and contributes significantly to the overall immersive experience of our project.

4.2 Software

In this section, we provide a comprehensive list of all the software required for our project. We detail the specific applications, their purposes and how they contribute to achieving the project's objectives.

4.2.1 SteamVR

SteamVR is a virtual reality platform developed by Valve Corporation that supports a wide range of VR hardware, including the HTC Vive Pro Eye. It is integrated with the Steam ecosystem, allowing users to access a vast library of VR games and applications

Figure 4.2: Tobii eye-tracking hardware inside HTC Vive Pro Eye.

Figure 4.3: Leap Motion hand tracking device.

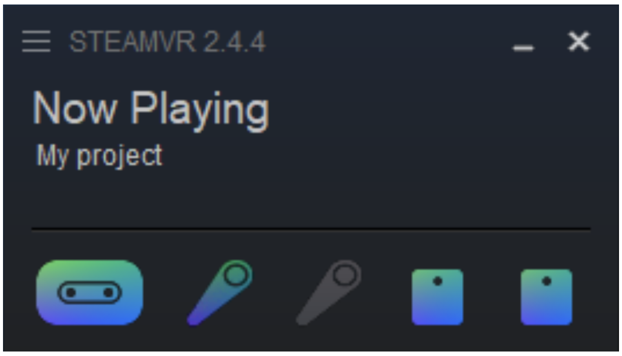
available on the Steam Store. Providing the necessary tools to set up, control and use VR headsets. It acts as a bridge between the VR hardware and applications, enabling a seamless VR experience (Fig. 4.4(a)).

4.2.2 VIVE Console for SteamVR

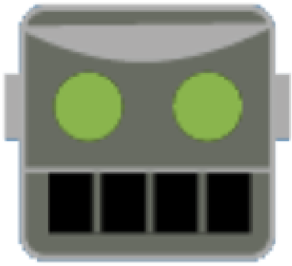
VIVE Console for SteamVR is software specifically designed for HTC Vive headsets. It allows users to configure and optimize their VR setup. Essential for setting up and running HTC Vive Pro Eye, mainly for enabling advanced features like eye tracking, hand-tracking and proper calibration of the device.

4.2.3 Unity Engine

Unity is a versatile and widely-used game engine developed by Unity Technologies, providing tools for creating 2D, 3D, VR and AR applications across multiple platforms



(a)



(b)

Figure 4.4: (a) Steam VR interface, (b) SR Runtime icon showing that eye tracking is enabled.

and various VR/AR devices. It features a user-friendly editor, supports C# scripting and offers real-time rendering with Universal and High Definition Render Pipelines for different performance needs. Unity's Asset Store supplies a vast array of assets and plugins, while its Animator and physics engines enable complex animations and realistic interactions. Popular for VR and AR development, Unity also facilitates team collaboration through Unity Teams and Cloud Build. Great documentation, tutorials and strong community support from developers, make Unity ideal for game development, simulations, architectural visualization and education. Additionally, it includes monetization and analytics tools, helping developers optimize and monetize their creations.

4.2.4 SRAnipal

SRAnipal is a software developed by HTC for their Vive VR headsets, designed to enhance the virtual reality experience through advanced eye-tracking and facial expression tracking. The core component of SRAnipal is its integration with the Vive Pro Eye and other compatible Vive headsets, enabling precise and real-time tracking of eye movements and facial expressions, which significantly elevates the level of immersion and interactivity in VR environments (Fig. 4.4(b)). SRAnipal uses advanced eye tracking technology to capture the user's gaze direction, pupil dilation and blink rate. Beyond eye movements, SRAnipal captures a wide range of facial expressions, including eyebrow raises, smiles, frowns and other subtle facial movements. This capability allows avatars in social VR applications and virtual meetings to reflect the user's real facial expressions, making interactions more natural and expressive. SRAnipal comes with a comprehensive SDK that allows developers to integrate eye and facial tracking functionalities into their applications. The SDK provides robust APIs and sample code, making it easier to implement advanced features such as gaze-based interactions, attention-aware interfaces and real-time avatar animation.

4.2.5 OpenXR

OpenXR is an open standard developed by the Khronos Group created to provide a unified framework for developing VR and AR applications. Designed to streamline and simplify the VR/AR ecosystem, OpenXR enables developers to write applications compatible across a wide range of hardware platforms and devices without needing to customize their code for each specific system.

4.2.6 Tobii XR SDK

It is software development kit (SDK) provided by Tobii, a company that created our eye-tracker. With the SDK, we can access precise eye tracking data to implement features such as foveated rendering. Which optimizes graphical performance by reducing the resolution in peripheral vision areas while maintaining high fidelity to places the user is looking at. Additionally, the SDK facilitates natural user interactions and deeper analytics by tracking where and how long users look at different elements within the virtual environment. It enabled us to integrate eye tracking functionality into our project in Unity, allowing for features like object highlighting by looking at them.

4.3 Environment

Our project is an extension of an existing Unity project[1] containing the simulated model of robot NICO (Fig. 1.2) in Unity. We made small changes in the provided environment to ensure that it is suitable for the requirements of our project.

NICO, like all elements in Unity, is composed of GameObjects that we can manipulate either directly or through scripts. This flexibility allows for great customization and control over the behaviours and interactions of these GameObjects. By leveraging Unity's scripting capabilities, which use the programming language C#, we can dynamically adjust the properties of our environment. This modular approach is needed to create a responsive and adaptable system within our project.

[1] `https://github.com/iveta331/NICO.git`

Chapter 5

Implementation

Our implementation enables NICO to look at or follow any GameObjects inside Unity. With the use of an eye-tracker, we can use our gaze to look at objects and NICO will respond similarly (Fig. 5.1). To compare different types of operation we implemented scripts for hand and controller tracking.

5.1 Gaze tracking

Here we describe the inner workings of scripts designed to enable object interaction based on eye tracking. It also includes a simplified diagram illustrating the workflow of the scripts (Fig. 5.2).

5.1.1 Receiver

The *PositionReceiver* (Alg.1) is our component responsible for managing the articulation of a head and neck in response to the position of a target object. This class is implemented as a Unity MonoBehaviour class, making it easy to attach to GameObjects in the Unity editor. Here's a detailed breakdown and explanation of each part of the script.

MonoBehaviour

This is a base class from which every script derives if it is to be attached to a GameObject. It provides the framework for creating components that can be added to objects and managed by Unity's scripting engine. This class provides various lifecycle methods that Unity calls at a specific time during execution. Some of these methods include:

- *Awake()* called when the script instance is being loaded,
- *Start()* called before the first frame update if the script instance is enabled,

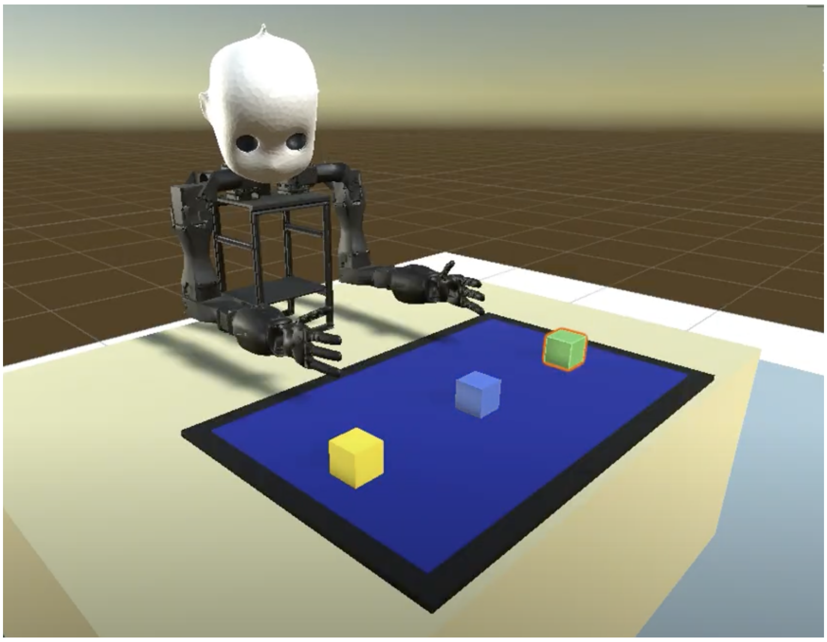


Figure 5.1: NICO focusing on the same object that we are observing, which is indicated by the object turning yellow in colour.

- *Update()* called every time a frame is loaded,
- *OnDestroy()* called when the MonoBehaviour will be destroyed.

It also provides a component system which can be used for interaction of scripts with other components attached to the same GameObject or other GameObjects.

Event system

This is a feature in Unity that allows for a flexible and decoupled way of handling events and callbacks. It is built on the concept of the observer, where one object maintains a list of its dependents and notifies them of any state changes, usually by calling one of their methods.

Under this class, we created our own *PositionEvent* that takes the coordinates of an object and sends it to the listener. We can add listeners using the Unity editor or write them in the code. Our listener is method *OnPositionReport* inside the *PositionReceiver* class. This enables data transmission directed towards the *PositionReceiver* from all GameObjects, enabling it to receive up-to-date positional information.

References

This is important for the next part. Interface in Unity allows the assignment of objects to variables inside the scripts. When we assign a GameObject, such as a Camera, to a public variable inside via the Inspector, Unity does not just store the name of the camera. Instead, it keeps a reference to the camera's instance. This is managed through Unity's object referencing system within its serialization framework.

Movement calculation

With reference to NICO's head, which is an ArticulationBody, the PositionReceiver performs vector calculations upon detecting a position update for a specific GameObject. These calculations are crucial for determining the accurate orientation and rotation of NICO's head. The computed results are then converted into angles. This conversion is necessary for utilizing Unity's ArticulationBody component, which facilitates complex physics-based simulations and movements, providing precise control over NICO's articulated body parts.

By continuously checking for position inputs from objects and applying real-time vector adjustments, the *PositionReceiver* ensures that NICO's head moves correctly. This makes so the robot look as if it is looking at an object.

Algorithm 1 Pseudocode of the *PositionReceiver* class

```
while not received any object do
    listen for objects
end while
if articulation not initialized then
    disable this script
end if
if object is not correctly received then
    disable this script
end if
if head already turned to object then
    disable this script
end if
movement ← calculate movement needed
move articulation correctly
```

5.1.2 Reporter

The *PositionReporter* class is a Unity MonoBehaviour class that interfaces with the Tobii Gaze-Focused Object Manipulation system. This class reports the position of a referenced GameObject when it is looked at and provides visual feedback by changing its colour. The class implements the IGazeFocusable interface, indicating that it can respond to gaze focus changes. It is designed to be a versatile and easily usable component that can be applied to any GameObject within Unity. Once attached and properly configured, the *PositionReporter* transmits the positional data of its associated GameObject to *PositionReceiver*(Fig. 5.2).

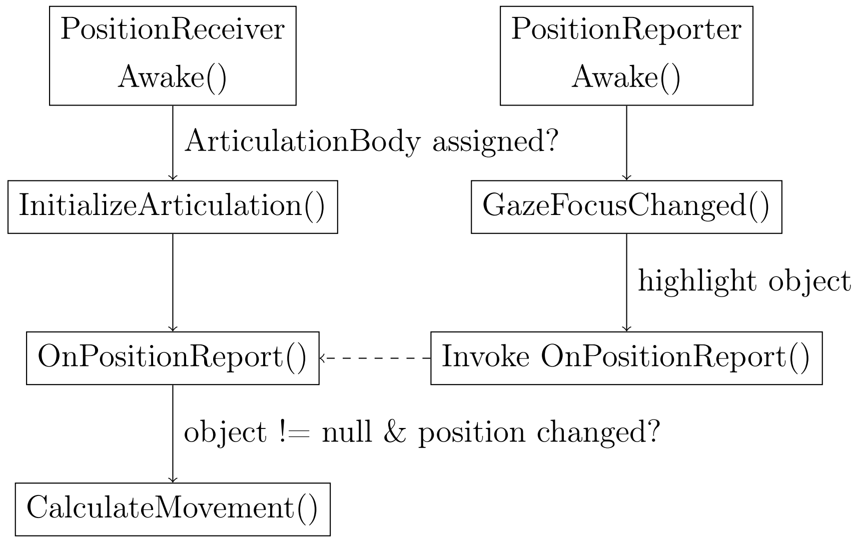


Figure 5.2: Workflow Chart for PositionReceiver and PositionReporter

IGazeFocusable

The interface provided by Tobii updates every frame and informs if the object the script is attached to gains or loses focus. Using the event created by us, we can send the position of the referenced object to listeners, in this case, the method inside *PositionReceiver* class.

5.1.3 Arm Extension

The *ArmExtend* and *EyeContact* features incorporate many of the principles discussed in Sections 5.1.1 and 5.1.2. By applying these principles, we have enabled functionality that allows NICO to perform specific actions based on our gaze. For instance, when we direct our gaze towards NICO's hand, he extends it, allowing us to place an object into it (Fig. 5.3). Additionally, NICO can retract his hand and establish eye contact with us when we look directly into his eyes, enhancing the sense of interaction.

5.1.4 Validation

In each script we implemented that depends on specific information about our Unity objects such as NICO's arms, his head, our position within the 3D scene and the cubes symbolizing our interactive objects, we include validation checks to ensure these objects are properly configured and present in the scene. By leveraging the properties of

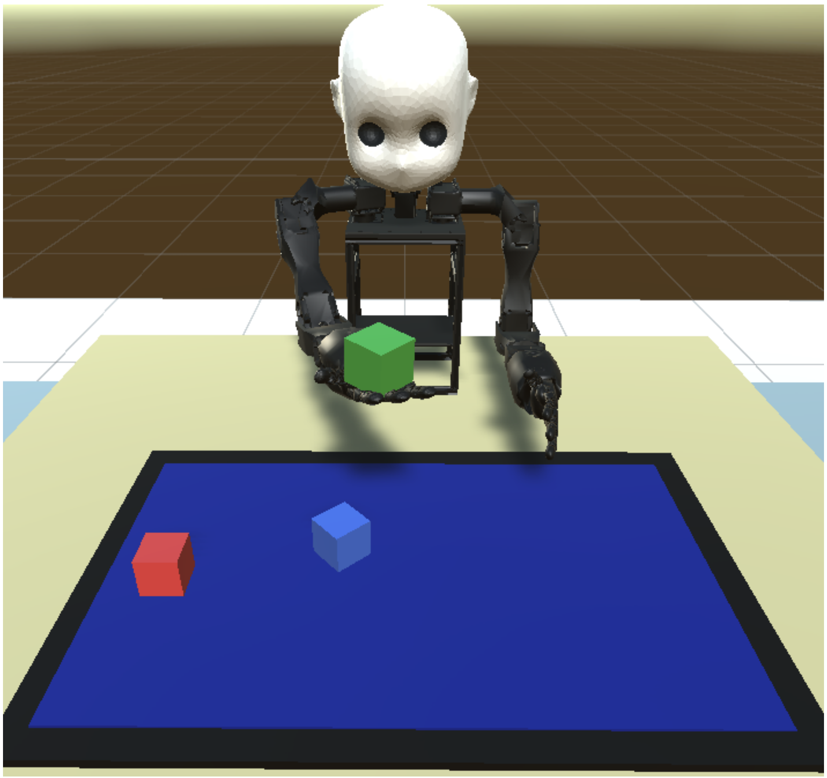


Figure 5.3: NICO has its arm extended and is holding one of our interactive objects.

the Unity environment, as detailed in Section 4.1.3, we verify whether the objects referenced in the script are still active within the scene and possess the correct attributes. This ensures that our scripts function correctly and that the interactions within the environment are reliable and consistent.

5.2 Controller and hand tracking

To enhance our ability to assess and compare various methods, we decided to incorporate both hand tracking and controller tracking. By adding these additional control mechanisms, we can more thoroughly evaluate and compare the effectiveness of different approaches to the interaction with the VR environment.

5.2.1 Hand tracking

As mentioned in Section 4.1.3 we utilized the Leap Motion tracker. The tracker uses a sensor to capture our hands. Each hand is represented as 26 points in 3D space as shown in Fig. 5.4. We mapped those points directly to our hand models in Unity, enabling us to create a real-time representation of hands within our VR scene. Since these hand models are GameObjects, we have complete control over their behavior and characteristics. This allows us to assign various properties to them, such as altering their shape, size and colour, to better suit the needs of our VR environment.

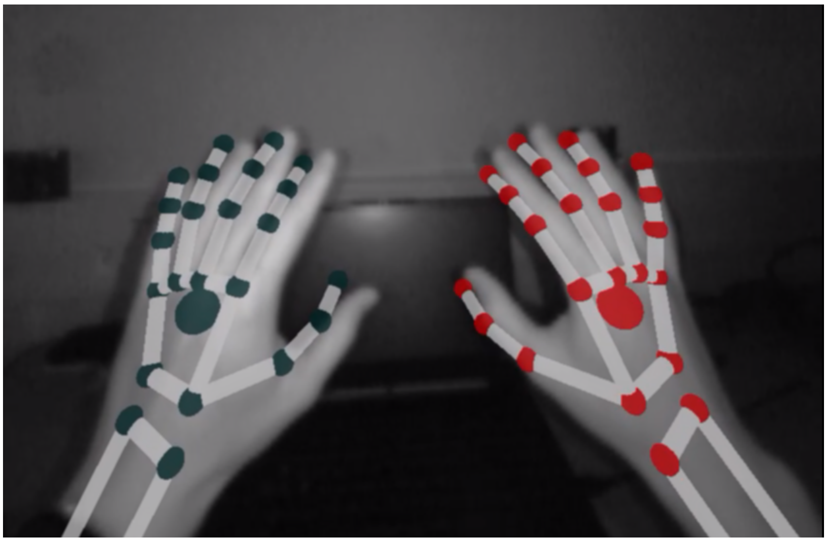


Figure 5.4: Screenshot from the camera during real-time hand tracking.

Additionally, we used a script that enhances the interaction experience by smoother grabbing of our interactive objects, making the overall interaction more intuitive and seamless. To include all of this within our scene we added a model of the hands and interaction manager specific to hand tracking. With some adjustments of interactions inside Unity, our hand models started to respond to information provided by trackers. With this, we can interact with the scene, grab objects and control the basic UI we created (Fig. 5.5).

Our UI required the implementation of scripts responsible for interaction with the environment. Since all objects in Unity are connected, allowing us to have control of the entire scene. First, we created a script to reset the main camera and cubes, our interactive objects, to their initial positions. Additionally, we added basic controls, such as a button in the UI, which allows us to turn off the scene. This UI is available only for interaction with hand tracking.

5.2.2 Controller tracking

We implemented controller tracking primarily to introduce an additional method of interaction with the scene. This approach has proven to be not only more reliable but also more accessible to a wider range of users. Our project is built on the open standard OpenXR, which ensures compatibility across various types of devices, allowing for a consistent user experience. While not every HMD includes integrated hand tracking and not everyone has access to standalone tracking devices, the majority of HMDs are equipped with some form of controller. This makes controller tracking a more universally available option for interaction. Furthermore, from a reliability standpoint,

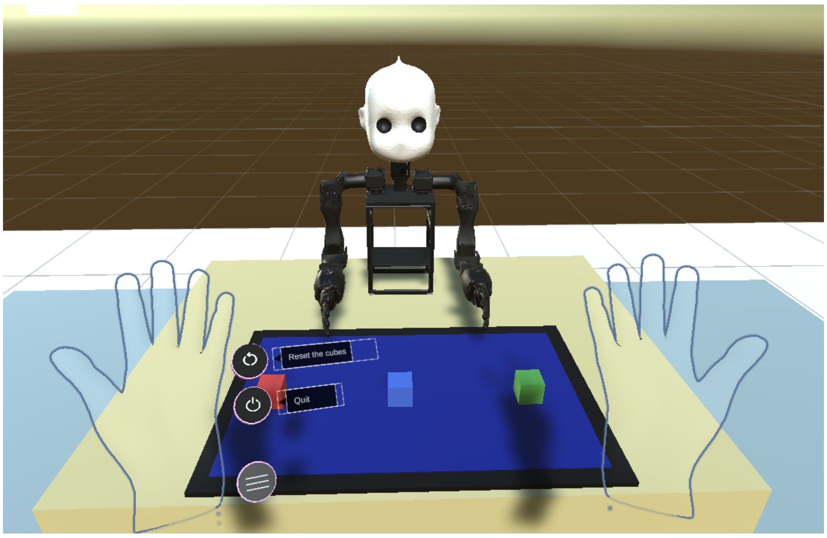


Figure 5.5: Screenshot from our scene A with hand models and UI for hand tracking.

controller tracking has demonstrated itself to be a more precise method of interaction. This precision is due to its reliance on various advanced tracking technologies, such as gyroscopes and infrared sensors, contributing to a more accurate and stable user experience.

To take advantage of the controllers, we implemented a script that maps certain controller buttons to specific functions we want to utilize for interaction within the scene. To enhance the immersive experience, we paired these functions with hand models that closely resemble those used in our hand tracking system. This visual consistency helps maintain a cohesive user experience across different interaction methods. For the various functions, we implemented the ability to grab objects and point at them, complete with corresponding animations. These animations are designed to make the interactions feel more natural and intuitive, further immersing users in the virtual environment.

LineConnector

We implemented the *LineConnector* script to enable pointing at interactive objects within our scene, aiming to create a functionality that closely mirrors our eye tracking interactions with objects. Since we use hand models to represent our controllers in Unity, we can generate a visual line extending from our fingertips, which helps to indicate the direction in which we are pointing.

When this line intersects with an interactive object in the scene, it triggers notifications to other objects, allowing them to respond accordingly. For instance, this could activate another script that causes NICO to look at the object being pointed at (Fig. 5.6).

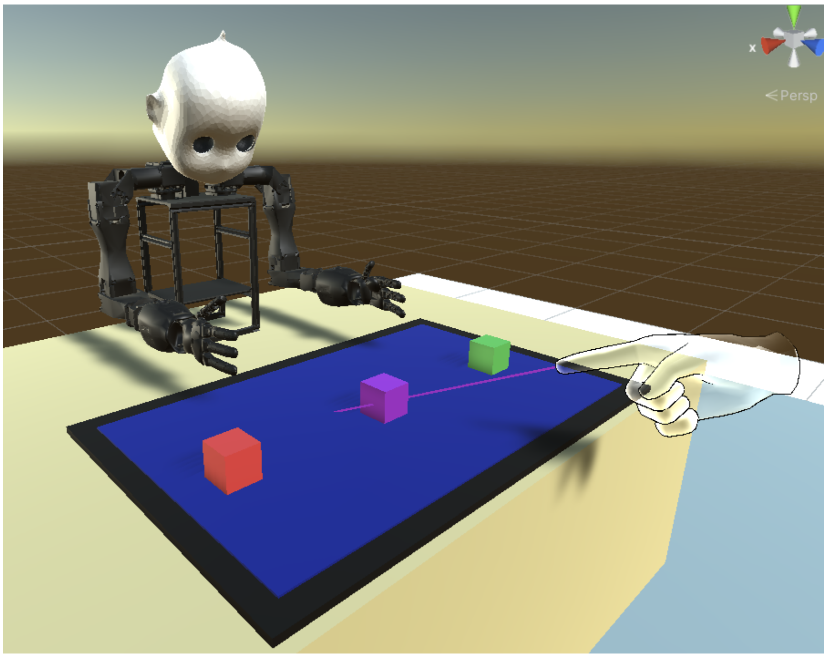


Figure 5.6: Screenshot from our scene showing model of hand representing controller pointing at the object.

5.3 Challenges

Despite the great developer support available for used hardware and software components in our project, several issues arose. For instance, we encountered instances of SRAnipal software failing to properly initialize the connection with the Tobii hardware. This malfunction prevented the calibration and usage of the eye tracker, even though the system appeared to be functioning correctly. In addition to these software-related problems, we also faced reliability issues with both OpenXR and SRAnipal. A specific bug within the OpenXR software occasionally prevented the correct initialization of SRAnipal, leading to the eye-tracking feature becoming non-operational once again. Later we managed to fix this using another software VIVE OpenXR Plugin provided by ViveSoftware. These software issues, however, were not the only obstacles we faced. Hardware-related problems also posed significant challenges. We frequently encountered connectivity issues between the HMD and the computer, which not only caused the eye-tracking functionality to fail completely but also disrupted other essential tracking functions, such as the accurate positioning of controllers and head tracking.

Additionally, there are certain inaccuracies to consider. For example, the technical specifications of the HTC VIVE Pro report a spatial accuracy of 0.5° to 1.1° within a 20° field of view. Such differences could lead to variations in results, highlighting the importance of accounting for these potential errors in our assessments.

Furthermore, the latency of both the software and hardware has proven to be critically important. Delays in processing or response times could significantly impact the accuracy, effectiveness and overall immersion of the system, proving the need for careful management of latency issues.

5.4 Testing the interaction components

The final part of our work, following the implementation of the components mentioned above, was testing our work. We had four respondents, respondent 2 had previous experience with both controllers and hand tracking. Respondents 1, 3 and 4 had similar experience with controllers and no experience with hand tracking.

5.4.1 Test description

For this purpose, we created two scenes. In scene A, the respondent stood in front of NICO with their hands close to its virtual body (Fig. 5.5). Scene B offered an elevated, angled view of the respondent above NICO, with the hands positioned further from the virtual body and not directly connected to it, as shown in Fig. 5.7. This setup allows for a wider range of motion since the virtual hands are not tied to the user's position in VR, making it more suitable for use in confined spaces. The controls of our module were properly explained to our respondents. After that, they were first placed in the scene where they stood in front of NICO. This view is more intuitive because it closely resembles the real world. After completing the testing scenario in this scene, they were then moved to scene B with the overhead view to perform the same tasks.

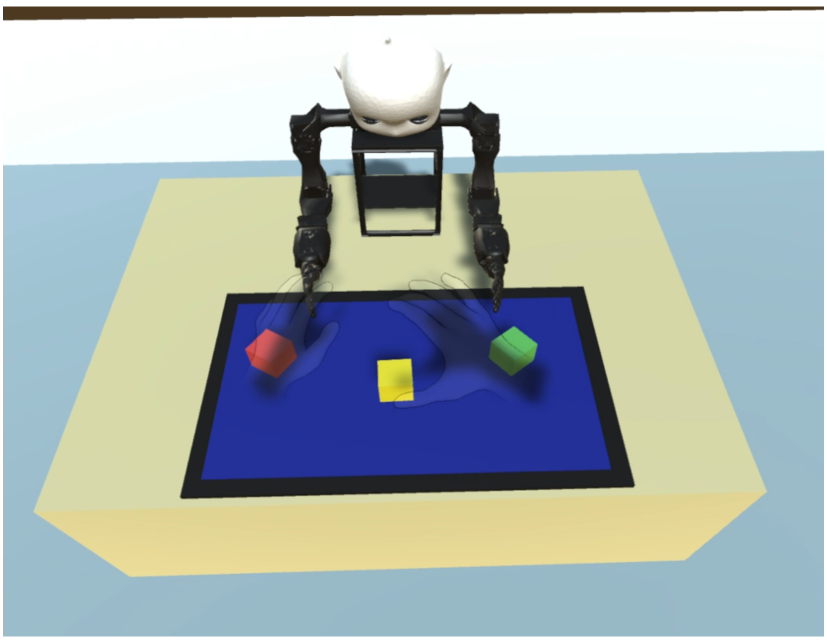


Figure 5.7: Screenshot from our scene B, providing an overhead view of NICO.

Our testing scenario consisted of two tasks. The first task involved handing a cube to NICO's extended hand. The second task required stacking objects, represented by cubes, on top of each other in front of the respondent. They had five attempts to complete each task, with a time limit set to thirty seconds. Although the tasks were relatively simple, the time limit helped us identify instances where the respondent failed the test. After completing the tests, the respondents rated their experience with our implementation in the categories of controllability and immersion. We used a scale from 1 to 5, where 1 was the lowest and 5 was the highest score. In the category of controllability, we assessed the subjective difficulty of adapting to an unfamiliar user interface its intuitiveness and how easily they could make moves they intended in VR. Next, we evaluated the level of immersion they experienced in VR environment while completing the tasks. This assessment focused on how deeply they felt engaged or absorbed in the VR setting and how closely it mimicked real-world interactions.

5.4.2 Test results

The next step is to evaluate the testing and the current state of the interface. In Tables 5.1 and 5.2, we can see the success rates of task execution by individual respondents and their ratings of the interface in the categories of immersion and controllability divided by type of control. For controller tracking, the average ratings for immersion and controllability were **3.75** and **4.5**, respectively. For hand tracking, the ratings were lower, with **3.25** for immersion and **2.5** for controllability.

Respondent	Scene A		Scene B		Sum	Immersion	Controllability
	Test 1	Test 2	Test 1	Test 2			
1	5	5	5	4	19	4	4
2	5	4	5	5	19	3	4
3	5	4	5	5	19	4	5
4	5	5	5	5	20	4	5

Table 5.1: Results of our test scenarios using controller tracking.

Respondent	Scene A		Scene B		Sum	Immersion	Controllability
	Test 1	Test 2	Test 1	Test 2			
1	3	4	5	2	14	4	3
2	4	4	4	4	16	4	4
3	4	3	4	3	14	2	1
4	5	2	4	1	12	3	2

Table 5.2: Results of our test scenarios using hand tracking.

5.4.3 Evaluation of testing

With the test results and verbal assessment from respondents, we came to these conclusions:

- Eye tracking provided a significant advantage in task immersion. However, in most of the requested scenarios, eye tracking alone did not add value. Nevertheless, there were no issues with the tracking itself.
- Interaction was better with the controllers compared to hand tracking. Controllers were easier to learn and control. In terms of controllability, they outperformed based on the evaluation and the number of tasks successfully completed. Even, the immersion rating was slightly higher.
- Hand tracking performed worse in both controllability and immersion. We expected immersion to be higher, but every one of our respondents encountered issues with hand tracking related to the precision and ability of the trackers to accurately follow movements. Frequent loss of tracking also negatively impacted the results. This is also reflected in the number of completed tasks compared to controller tracking.
- Accurate tracking of every type of interaction is crucial for both immersion and controllability. Hand tracking encountered the most issues with movement tracking. However, controllers also had some issues and there were instances where even the HMD lost its tracking.

Some of the issues with hand tracking were caused by general inexperience with the technology. However, from the tasks successfully completed by each respondent, we observed that this had a smaller impact than we initially assumed. The recorded testing scenarios clearly show that each respondent had a different approach to interacting with objects. Some of these approaches caused the hand models in Unity to lose track or start shaking, making the tasks very difficult to complete. Unlike controller tracking, which was mainly affected by a lack of experience and precision issues, hand tracking introduced a different problem significant enough to lose out to controller tracking, even in terms of immersion. This problem mainly occurred in scenes where the respondents were performing tasks with a view from above, but it was also present in the scene facing the front of NICO, specifically affecting the left hand of respondents 3 and 4.

The assessments from the respondents, along with the small differences between the experienced respondent and others, indicated that the interaction system is very intuitive and easy to use. Given this and the high success rate of both easier and more challenging tasks—with only hand tracking failing in specific cases—we assess the interaction components as beneficial for engaging with NICO in VR, especially since eye tracking did not fail once.

Chapter 6

Conclusion

In this thesis, we explored the integration of human eye gaze following in VR environments to enhance HRI. The main objective was to enable a virtual robot, NICO, to perceive and respond to human gaze and other interaction methods, primarily using eye-tracking technology, thereby creating a more intuitive and engaging experience.

Our approach involved the preparation of the Unity-based VR environment equipped with the HTC Vive Pro Eye headset and implementing eye tracking, hand tracking, and controller tracking functionalities using compatible software. We developed a module that allowed the NICO robot to follow human eye gaze, along with modules for controlling the scene using hands and controllers. We also demonstrated the potential for more natural and responsive HRI in virtual settings, highlighting several key benefits of using eye gaze tracking in VR for HRI, such as improved user engagement and more precise control over the environment.

While this thesis primarily focused on VR within the Unity framework, the work could also be extended to AR. Using Unity's AR capabilities and real object tracking, allowing interaction with NICO in AR or the physical world.

However, we also identified significant challenges, particularly related to hardware calibration, software latency, and the accuracy of input tracking. Some of these problems could be resolved with better software implementations, while others are tied to the specific hardware we are using.

In conclusion, while the integration of eye gaze, hand, or controller tracking in VR presents substantial opportunities for advancing HRI, it also requires careful consideration of technical limitations and ongoing refinement of both hardware and software components.

Bibliography

Adhanom, I. B., MacNeilage, P., and Folmer, E. (2023). Eye Tracking in Virtual Reality: a Broad Review of Applications and Challenges. *Virtual Reality*, pages 1–24.

Admoni, H. and Scassellati, B. (2017). Social Eye Gaze in Human–Robot Interaction: A Review. *Journal of Human-Robot Interaction*, 6(1):25–63.

Bastani, B., Turner, E., Vieri, C., Jiang, H., Funt, B., and Balram, N. (2017). Foveated Pipeline for AR/VR Head-Mounted Displays. *Information Display*, 33(6):14–35.

Breazeal, C. (2004). Social interactions in HRI: the robot view. *IEEE Transactions on Systems, Man, and Cybernetics, part C (applications and reviews)*, 34(2):181–186.

Bruno, R. R., Wolff, G., Wernly, B., Masyuk, M., Piayda, K., Leaver, S., Erkens, R., Oehler, D., Afzal, S., Heidari, H., et al. (2022). Virtual and augmented reality in critical care medicine: the patient's, clinician's, and researcher's perspective. *Critical Care*, 26(1):326.

Clay, V., König, P., and Koenig, S. (2019). Eye Tracking in Virtual Reality. *Journal of Eye Movement Research*, 12(1).

Dafarra, S. (2020). *Predictive Whole-Body Control of Humanoid Robot Locomotion.*

Evan-Amos (2017). The oculus rift cv1 (consumer version 1), a virtual reality headset made by oculus vr and released in 2016. `https://commons.wikimedia.org/w/index.php?curid=62084228`. Accessed: 2024-07-04.

Farnsworth, B. (2018). Eye Tracking: The Complete Pocket Guide. *Imotion. December 4th.*

Hagis, C. (2003). History of Robots. *Pridobljeno iz https://citeseerx. ist. psu. edu/viewdoc/download.*

Hroboň, M. (2021). Interakcia človek-robot pomocou gest. Master's thesis, Slovenská technická univerzita v Bratislave FEI ÚRK.

Jerald, J. (2015). *The VR book: Human-Centered Design for Virtual Reality*. Morgan & Claypool.

Kerzel, M., Strahl, E., Magg, S., Navarro-Guerrero, N., Heinrich, S., and Wermter, S. (2017). NICO - Neuro-Inspired COmpanion: A developmental humanoid robot platform for multimodal interaction. In *Proceedings of the IEEE Int. Symposium on Robot and Human Interactive Communication*, pages 113–120.

Kowler, E. (2011). Eye Movements: The Past 25 Years. *Vision Research*, 51(13):1457–1483.

Kranen, H. (2024). Apple vision pro with the solo knit band option. `https://commons.wikimedia.org/w/index.php?curid=147747709`. Accessed: 2024-07-04.

Kurfess, T. R. et al. (2005). *Robotics and Automation Handbook*, volume 414. CRC press Boca Raton, FL.

Rama (2019). An image of the robot kismet. `https://commons.wikimedia.org/w/index.php?curid=89032593`. Accessed: 2024-06-04.

Richard, B. (2021). Live virtual eeality eye-tracking data visualization. Technical report.

Romdhane, L. and Zeghloul, S. (2009). Al-jazari (1136–1206). In *Distinguished Figures in Mechanism and Machine Science: Their Contributions and Legacies, Part 2*, pages 1–21. Springer.

Russell, B. (2017). *Robots: The 500-year Quest to Make Machines Human*. Scala Arts Publishers, Inc.

Sheridan, T. B. (2016). Human–robot interaction: status and challenges. *Human Factors*, 58(4):525–532.

Sipatchin, A., Wahl, S., and Rifai, K. (2021). Eye-tracking for clinical ophthalmology with virtual reality: A case study of the htc vive pro eye's usability. In *Healthcare*, volume 9, page 180. Mdpi.

Voštinár, P., Horváthová, D., Mitter, M., and Bako, M. (2021). The look at the various uses of VR. *Open Computer Science*, 11(1):241–250.

Wang, Y. and Li, X. (2024). Eye Movement Tracking in Ocular, Neurological, and Mental Diseases. *Frontiers in Neuroscience*, 18:1364078.

Weier, M., Stengel, M., Roth, T., Didyk, P., Eisemann, E., Eisemann, M., Grogorick, S., Hinkenjann, A., Kruijff, E., Magnor, M., et al. (2017). Perception-driven accelerated rendering. In *Computer Graphics Forum*, volume 36, pages 611–643. Wiley Online Library.

Čorňák, M. (2020). Interakcia človek-robot pomocou gest. Master's thesis, Slovenská technická univerzita v Bratislave FEI ÚRK.

Appendix A: Source code and testing data

All files are available at `https://github.com/luboshellesch/NICO-Eye-Tracking.git`. One can find the specific parts of the code mentioned in Implementation in file located at *NICO-Eye-Tracking/Assets/Scripts*.

To run our program, begin by downloading the zip file from the site mentioned above. While it is supposed to work on any HMD supported by OpenXR however we recommend the hardware specified in the Hardware section. Especially the HTC Vive Pro Eye to use its eye tracking capabilities. The next step is to download Steam VR from Steam, Unity Hub, and Vive Console, which should include SRAnipal, from their respective sites. Additionally, download the hand tracking software from UltraLeap. Follow all instructions and once all the software is installed, connect and set up your HTC Vive Pro Eye. Run SR Runtime and Steam VR. If everything is functioning correctly, you can run our program from Unity Hub. In the Unity interface, navigate to the Scenes folder, locate myNicoScene, and double-click it. Finally, run the game from the top of the interface. We provide more detailed installation guide on our website mentioned at the start.

The data from testing are available on at `https://drive.google.com/drive/folders/1Ll1QZbo4ACzrwPIX5CMpNOF2JKLEmkxl?usp=sharing`.