



SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY
INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics: Games Engineering

**Integration of Mixed Reality and Touchscreen
Interfaces for Humanoid Robot Embodiment in
a Virtual Clinical Setting**

Yinfeng Yu





SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY
INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics: Games Engineering

Integration of Mixed Reality and Touchscreen Interfaces for Humanoid Robot Embodiment in a Virtual Clinical Setting

Integration von Mixed Reality- und Touchscreen-Schnittstellen für die Verkörperung humanoider Roboter in einer virtuellen klinischen Umgebung

Author: Yinfeng Yu
Supervisor: Prof. Gudrun Klinker
Advisor: Christian Eichhorn
Submission Date: 15.11.2023



I confirm that this master's thesis in informatics: games engineering is my own work and I have documented all sources and material used.

Munich, 15.11.2023

Yinfeng Yu

Acknowledgments

This paper is by no means only my own achievement. Over these six months of work, there have been many people who have stood by my side and provided me with their help.

First and foremost, I would like to express my deepest gratitude to my advisor, Mr. Christian Eichhorn. It is he who gave me the opportunity to get in touch with and understand the intriguing ReduSys project. Without his patient guidance and invaluable advice, this thesis would not exist. I have learned a lot from him, and I feel incredibly fortunate to have completed the final chapter of my master's journey.

Secondly, I want to thank Kawabata and Gao. Their outstanding work forms the cornerstone of most of what I have done. I can hardly imagine how lost I would be without their contributions when I stand before all of this.

Also, I would like to thank Devanthro, especially Ms. Alona Kharchenko. Her advice, guidance, and assistance have allowed me to establish such a strong and unforgettable connection with a such cutting-edge robot.

Furthermore, many thanks to all the people who participated in my user study. I am deeply appreciative of the time, patience, and kindness you have shown.

Last but not least, I want to thank my family for their care, encouragement, and support. I want to thank the girl who has always been by my side, who is the most adorable and reliable companion in my life and will continue to be. I want to thank my friends for making me feel just as cherished as I do even when I am thousands of miles away from home. I love you all.

Abstract

This paper aims to develop a comprehensive Robody control platform that integrates a smartphone and an augmented reality (AR) head-mounted display (HMD) to explore new modes of achieving remote embodiment through Robody. We created a multifunctional smartphone application allowing users to remotely monitor Robody, assign autonomous tasks, and use the smartphone as an alternative to AR controllers. Additionally, we proposed two smartphone-based and one hand tracking-based methods for controlling Robody's hands. We successfully integrated our developed control methods into the physical Robody, validating the feasibility of our approach. We evaluated the usability, embodiment, and performance of various control methods through a user study. The results indicate that our control system's usability surpasses the average level and induces a certain level of embodiment. In the experiment, the hand tracking control mode performed the best, while the smartphone pointer control mode performed the least satisfying. Our work demonstrates the potential of combining smartphone and AR HMD, as well as hand tracking-based control methods in Robody control, providing a foundation for achieving further embodiment and improved control in the future.

Contents

Acknowledgments	iii
Abstract	iv
1. Introduction	1
2. Related Work	5
2.1. Robots in Healthcare	5
2.1.1. Overview	5
2.1.2. Assistive Robots	6
2.1.3. Humanoid Robots	7
2.1.4. Existing Robots in Healthcare	8
2.1.5. Ethical Concerns	10
2.2. Use of Simulation in Robotics	11
2.2.1. Motivation	11
2.2.2. Limitations	13
2.3. Digital Twin	13
2.3.1. Definitions and Concepts	13
2.3.2. Advantages	14
2.3.3. Classification	16
2.3.4. Challenges	17
2.4. Interaction with Smartphone	18
2.4.1. Remote Interaction	18
2.4.2. 3D Interaction	20
2.5. Human-Robot Interaction (HRI)	21
2.5.1. HRI Styles	22
2.5.2. Challenges in Robot Teleoperation	23
2.5.3. Virtual Reality (VR) in HRI	25
2.5.4. Mixed Reality (MR)/Augmented Reality (AR) in HRI	27
2.5.5. Touchscreen/Mobile Phone/Smartphone in HRI	28
2.5.6. Hand Tracking in HRI	31
2.6. Embodiment	32
2.6.1. Definitions and Concepts	32
2.6.2. Embodiment in Robotics and HRI	33
2.6.3. Experimental Studies	33

2.7.	Immersion and Presence	34
2.7.1.	Definitions and Concepts	34
2.7.2.	Factors Affecting Immersion	35
2.7.3.	Separation of Immersion and Presence	36
3.	Approach	38
3.1.	Introduction	38
3.1.1.	Overview	38
3.1.2.	Previous Works	38
3.1.3.	Goals	39
3.1.4.	Hand Control Strategies for Robody	41
3.2.	Tools and Technology	43
3.2.1.	Hardware	43
3.2.2.	Software	44
3.3.	Implementation	46
3.3.1.	Project Setup	46
3.3.2.	Device Communication	47
3.3.3.	User Interface	48
3.3.4.	Workflow	51
3.3.5.	Monitoring	52
3.3.6.	Task Assignment	54
3.3.7.	Hand Control	55
3.3.8.	Finger Control	62
3.3.9.	Displacement	65
4.	Evaluation	67
4.1.	Introduction	67
4.1.1.	Overview	67
4.1.2.	Goals	67
4.2.	Experimental Setup	68
4.2.1.	Integration with Physical Robody	68
4.2.2.	Tasks in the Virtual Environment	69
4.3.	User Study	71
4.3.1.	Participant Consent	71
4.3.2.	Experiment	71
4.3.3.	Questionnaire	72
4.4.	Results	75
4.4.1.	Integration with Physical Robody	75
4.4.2.	User Study Demographics	75
4.4.3.	System Usability	76
4.4.4.	Embodiment	77
4.4.5.	Project Specific Issues	79
4.4.6.	Task Complete Time	81

4.4.7. Discussion	84
5. Future Work	86
5.1. Further Application Development	86
5.1.1. Customized Control Interfaces	86
5.1.2. Multi-Platform Support	86
5.1.3. Known Issues	87
5.2. Better Hand Tracking Solution	87
5.3. Better Smartphone Control Method	88
5.4. Voice control	88
5.5. Integration with Physical Robody	88
6. Conclusion	89
A. Appendix	90
A.1. Consent Form	90
A.2. Questionnaire 1 (System Usability Scale)	91
A.3. Questionnaire 2 (Embodiment Questionnaire)	92
A.4. Questionnaire 3	94
List of Figures	96
List of Tables	98
Glossary	99
Bibliography	100

1. Introduction

Coronavirus disease 2019 (COVID-19¹) is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) virus. The first cases were first reported in China in December 2019 and rapidly spread across the world. This led World Health Organization (WHO) to declare a Public Health Emergency of International Concern on 30 January 2020, and to declare a pandemic on 11 March 2020². As of March 2023, variants of the disease have killed nearly 7 million people worldwide³. The United Nations Secretary General warned in 2020, that the coronavirus outbreak was the biggest challenge for the world since World War Two⁴.

In the face of such a highly contagious infectious disease, healthcare workers bear the brunt. As mentioned by Karlsson et al. in 2020, there have been many studies that have proved that healthcare workers are the group with the highest risk of infection. A cohort study of healthcare workforce across Scotland found that patient-facing workers were three times more likely to be infected with COVID-19 than non-patient-facing workers [1]. A survey of 595 healthcare workers in Italy from March to April 2020 showed that more than half of the infected healthcare workers had respiratory symptoms, and the chances of having anosmia and dysgeusia were much higher than those without infection. Mental health problems and poor sleep quality were also reported by a significant number of healthcare workers. Only less than one-third of the workers experienced no symptoms [2]. As of May 8, 2020, there were 152,888 reported cases of infection among healthcare workers worldwide, including 1,413 deaths [3], accounting for about 4% of the infected population by that time (over 3,8 million reported cases around the world in total³).

In addition to the occupational exposure risks of COVID-19 that can lead to illness and death, healthcare workers also face other occupational risks during high-intensity work in the pandemic. These risks further include: “skin disorders and heat stress from prolonged use of personal protective equipment (PPE), exposures to toxins because of increased use of disinfectants, psychological distress, chronic fatigue; and stigma, discrimination, physical and psychological violence and harassment”. These occupational health problems not only make healthcare workers experience various COVID-19 symptoms, but also suffer from more work-related illness, leading to “high rates of absenteeism, reduced productivity and

¹COVID-19 for Coronavirus disease 2019 is used in this thesis. Novel Coronavirus(2019-nCoV) Situation Report – 22. Feb. 2020. <https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200211-sitrep-22-ncov.pdf>.

²Coronavirus disease (COVID-19) pandemic. <https://www.who.int/europe/emergencies/situations/covid-19>.

³WHO Coronavirus (COVID-19) Dashboard. <https://covid19.who.int>.

⁴Coronavirus: Greatest test since World War Two, says UN chief. Apr. 2020. <https://www.bbc.com/news/world-52114829>.

diminished quality of care”⁵.

Caring for patients with infectious diseases has become a major challenge during the COVID-19 pandemic, especially given the high risk of occupational exposure faced by healthcare workers. First of all, entering and exiting infectious wards is a major inconvenience. According to the WHO interim guidance of 2021 July, healthcare workers should wear medical masks/respirators, eye protection (goggles or face shield), long-sleeved gowns, and medical gloves to ensure maximum protection when in contact with COVID-19 patients⁶. However, putting on and taking off a great number of PPE and performing disinfection procedures before and after visits to the wards can be very time-consuming and exhausting. Moreover, many non-specialized nursing tasks (such as passing items) do not actually require professionals to do them themselves, but their workload is further increased due to the additional protection procedure (compared to caring for non-infectious patients). Second, even when wearing the full set of PPE properly according to the guidance, there is still a risk of infection for healthcare workers around the infectious wards. Accidental contact with contaminants, as well as human errors, irregular operations, and accidents when putting on and taking off PPE, can all lead to occupational exposure. As mentioned above, prolonged use of PPE also brings additional symptoms to healthcare workers.

Considering these reasons, Devanthro⁷, Technical University of Munich⁸, Schön Klinik⁹, and others, initiated a collaborative project named ReduSys. One of the primary objectives of this project is to reduce unnecessary physical contact between healthcare workers and patients, while simultaneously promoting good contacts¹⁰. This initiative aims to improve patient care and the working environment for healthcare professionals, with a focus on reducing the risk of infection among healthcare workers and alleviating their workload.

The project allows for remote control by an operator to manipulate a humanoid robot named Robody. Robody is deployed as a substitute for healthcare worker, particularly in cases involving patients with highly contagious diseases, such as COVID-19. Through manual control by the operator, Robody is capable of performing tasks that are either not able or not suitable for autonomous execution by the robot itself.

However, controlling a robot often requires operators to have received relevant training. Healthcare workers without specific training in robot operation may struggle to effectively control the robot. One solution is to provide an immersive interacting experience that allows users to feel as they are the robot itself, a concept known as telepresence or remote embodiment. In this approach, users can control the robot’s arms as if they were using their own arms and perceive the robot’s environment from the robot’s perspective, even if they are physically located elsewhere. This concept of remote embodiment not only has applications

⁵COVID-19: Occupational health and safety for health workers. Feb. 2021. https://www.who.int/publications/i/item/WHO-2019-nCoV-HCW_advice-2021-1.

⁶Infection prevention and control during health care when coronavirus disease (COVID-19) is suspected or confirmed Interim guidance 12 July 2021. <https://www.who.int/publications/i/item/WHO-2019-nCoV-IPC-2021.1>.

⁷Devanthro. <https://www.devanthro.com>

⁸Technische Universität München. <https://www.tum.de>

⁹Schön Klinik. <https://www.schoen-klinik.de>

¹⁰Redusys. <https://www.redusys.de>

in the healthcare field, but can also extend to various aspects of daily life, such as remote visits and inspections. While these applications are intriguing, they fall outside the scope of the discussion in this thesis.

Previous work has explored solutions using virtual reality (VR) for the control described above. While VR head-mounted-display (HMD) can offer an excellent sense of immersion, using VR devices comes with numerous inconveniences and limitations. First, wearing a VR headset is a relatively complex and time-consuming setup process. This setup overhead can sometimes take more time than simple tasks such as bringing a bottle of water. Second, VR devices need to be used within specific safe environments. Healthcare workers or operators must be physically present at the location where the VR setup is, which can be inconvenient and inefficient. Moreover, when users wear a VR HMD, their field of view to the real world is entirely blocked, preventing them from engaging in other activities or maintaining awareness of their surroundings.

One solution to enhance flexibility is to replace VR HMDs with AR HMDs. AR HMD, with its see-through display, allows users to wear it for extended periods without significantly affecting other activities when not controlling Robody. The ability to wear them at any time means that users can engage in control operations whenever needed without the need for additional preparations.

The solutions mentioned above, using VR or AR, rely on specialized controllers to control Robody's hands. However, this introduces new challenges: specialized controllers are not practical for users to carry around. They served for a single purpose: only useful when controlling the robot and have little use when not controlling. Furthermore, controllers come in different shapes, making them less suitable for easy portability. An alternative solution need to be found.

Our project takes inspiration from the AR HMD-based solution proposed by Kawabata and Gao. We developed a novel control method that allows users to interact with and control Robody's arms using a touchscreen interface (smartphone), in the absence of AR controllers. Additionally, we introduced a hand tracking-based control method for Robody's hands, allowing users to control Robody's hands by simply moving their own hands in front of the camera.

Recognizing that not all tasks require manual intervention by operators, we integrated features such as task assignment and monitoring into the smartphone application. This allows Robody to perform certain low-risk, time-consuming tasks autonomously, like large-range displacement, thus relieving the burden on the users. In our system, the smartphone application acts as a multifunctional platform that integrates both touchscreen and AR interfaces.

We built a virtual clinical setting that simulates what Robody perceives in a real-world clinical environment. Inside this scene, we also built the digital twin of Robody whose capabilities mirror those of the physical Robody. Users wearing an AR HMD can see the simulated environment as if they were the Robody in that environment. They can also choose different control methods using a smartphone to interact with Robody.

Our implementation focuses on enabling reliable control and maximizing embodiment

using only a smartphone and users' hands. In the latter part of our work, we explored integrating this AR-based control method with the physical Robody and conducted a series of user studies to assess the performance of different control methods and gather insights into users' experiences and embodiment when using this control system.

2. Related Work

In this chapter, we have thoroughly examined various relevant domains related to our development and research. First, we guided the readers through the applications of robots in the healthcare industry, followed by a review of simulation and digital twin technologies involved in robotics. We then focused on investigating the human-robot interaction (HRI) aspect and delved into various interaction modes, along with relevant precedents that could assist our work. In the final sections, we investigated the concept of embodiment, immersion, and presence. At the end of each chapter, a tabular summarized the content of the chapter.

2.1. Robots in Healthcare

2.1.1. Overview

The use of robots in healthcare is not new. As early as 1985, the first documented case of robot-assisted surgery was reported, as mentioned by Kyrarini et al. [4] However, a 2017 study noted that while robotics has been extensively studied in industrial settings, there is limited research on service robots as assistants to humans [5]. In recent years, with advancements in robotics and artificial intelligence (AI), robots are increasingly being used alongside humans to provide assistance. In addition to technological developments, the increasing demand for nursing care due to the aging population and the shortage of healthcare workers are also important factors driving the development of robots in healthcare [4]. According to WHO statistics, in 2019, there were 1 billion people over the age of 60 around the world, and this number is projected to increase to 1.4 billion by 2030 and 2.1 billion by 2050 ¹. Additionally, a large population also faces difficulties with activities of daily living (ADL) and cognitive functioning tasks. Moreover, the shortage of healthcare workers makes it challenging to maintain a high standard of care and harder to get [6]. The introduction of healthcare robots can alleviate the impact of healthcare worker shortages and help maintain the quality of care by offloading many of the duties of healthcare workers, particularly in the face of predictable increases in demand for care [4].

Robots in healthcare can be classified into various types based on different criteria. They can be categorized as care robots, assistive robots, hospital robots, rehabilitation robots [4], based on their specific use. Among these, assistive robots can further be divided into physically-assistive robots (PAR) and socially-assistive robots (SAR), depending on how they provide assistance to people. The robots can also be classified as humanoid robots or non-humanoid robots based on their shape, and as teleoperated robots or autonomous robots based on their

¹Ageing. <https://www.who.int/health-topics/ageing>.

degree of autonomy.

The classification of robots according to use may sometimes be ambiguous. As discussed by van Wynsberghe in 2012, a care robot does not possess unique capabilities that define it, but is rather defined by how it is used [7]. The services provided by care robots, hospital robots and rehabilitation robots can be regarded as assistance to care receiver and/or healthcare workers, making the term “assistive robot” inclusive. In order to avoid confusion, in this thesis, we will not strictly differentiate between care robots, assistive robots, hospital robots, rehabilitation robots, etc., but rather refer to them collectively as assistive robots for discussion.

However, this section will not focus on the development and applications of surgical robots, as it is not relevant to the topic and scenarios we are concerned about. Instead, the focus will primarily be on robots that assist healthcare workers in their day-to-day care duties.

2.1.2. Assistive Robots

As mentioned earlier, defining care robots is challenging due to the lack of unique characteristics, as discussed by van Wynsberghe [7]. In terms of the way they are named, assistive robots are considered as robots that provide support and assistance to patients and healthcare workers in providing care. Feil-Seifer and Matarić’s work summarized that assistive robots include rehabilitation robots, wheelchair robots and other mobility aids, companion robots, manipulator arms for the physically disabled, and educational robots. In the past, the term “assistive robots” primarily referred to robots that assist people through physical interactions, but it has now been expanded to include robots that assist people through non-contact interactions [8]. Therefore, assistive robots can be categorized into two types based on the form of assistance they provide: physically-assistive robot (PAR) and socially-assistive robot (SAR) [9]. SARs are robots that offer support through social interaction to facilitate rehabilitation, learning, and recovery processes [8]. On the other hand, PARs are robots that provide assistance through physical interaction, including aiding users with activities like eating, dressing, and grooming [10, 11]. These robots have varying degrees of autonomy, ranging from fully autonomous to fully teleoperated, with many modern systems utilizing a hybrid autonomous approach [6].

According to the summary by Kyrarini et al., current applications of assistive robots are predominantly focused on the elderly, patients, and disabled individuals, with the aim of providing [4]:

1. Mental monitoring and assistance, such as reminders, emotional support, and motivation.
2. Physical assistance, including delivering and transporting items, as well as supporting ADL.
3. Diagnosis and assistance in the education of children with mental disorders, such as autism spectrum disorder (ASD).

The Robody robot involved in our project not only has the capability to socially assist patients, but also features articulated hands with grasping functionality and mobility, enabling

physical assistance to patients. Therefore, in our development process, we payed close attention to the implementation of its physical functionalities.

2.1.3. Humanoid Robots

Robots come in various forms. Influenced by science fiction (sci-fi) works like the famous *The Terminator*, people often associate the term "robot" with humanoid robots [12] [13]. However, the optimal form of a robot is determined by its intended functions. Different needs require specific forms of robots [13]. Based on morphology, robots can be classified into categories such as humanoid robots, animoid (animal-like) robots, machine-like robots, pet-like robots, screen-only robots, or a combination of these types [14]. In this thesis, we refrain from extensively discussing non-humanoid robots, as our work solely involves Robody, which is a humanoid robot equipped with human-like facial expressions and bodily capabilities.

As the name suggests, a humanoid robot is a robot that resembles the shape of a human being in its appearance. According to the summary of Kajita et al. [13], humanoid robots should possess three characteristics:

1. **Humanoid robots can operate in environments designed for humans.** The modern societal environment is designed with human beings in mind, including factors such as corridor width, stair height, and handrail placement that are tailored to human size and movements. When a robot has a human-like shape and behavior, it eliminates the need to modify the human environment to accommodate the robot's work. For instance, a wheeled robot may face challenges in navigating uneven terrains or stairs, which are easily accessible for healthy humans.
2. **Humanoid robots are capable of utilizing tools and objects designed for humans.** This is an advantageous characteristic as the last one. Many tools and objects are designed with human ergonomics in mind, such as screwdrivers and scissors that work optimally with articulated finger structures resembling those of humans. This eliminates the need for redesigning tools specifically for robots, making it more cost-effective.
3. **Humanoid robots has a human-like appearance.** When robots have a human-like appearance, people find it easier to personalize and interact with them. This is the main reason why humanoid robots are commonly depicted in human shape in sci-fi literature and media.

Extensive research has been conducted on the third feature mentioned above, which pertains to how the human-like appearance of robots can improve their social acceptability and facilitate natural interactions with humans. As robots are increasingly being deployed in everyday environments, they need to possess effective human-robot interaction (HRI) capabilities and garner human acceptance. Designing robots with these considerations in mind is crucial [15]. One approach to enhance the acceptance of robots and promote social interaction is by incorporating anthropomorphic (human-like) designs and "human social" features into robots. These may encompass human-like shapes, facial expressions, and natural human-like communication and interaction [15] [16].

Humanoid robots may not always evoke positive perceptions in humans. As depicted in Isaac Asimov's Robot Series and films like *The Terminator*, humanoid robots are often portrayed as posing a threat to human existence rather than being helpful assistants. These sci-fi works, along with media portrayal, have contributed to the formation of negative perceptions towards humanoid robots in the minds of many people [12].

2.1.4. Existing Robots in Healthcare

Prior to Robody, numerous humanoid robots designed for the healthcare industry had already been deployed. A considerable portion of these robots has been successfully mass-produced and commercialized. Apart from their application in human assistance, they are frequently utilized for research purposes as well.

Pepper

Pepper from SoftBank Robotics (formerly Aldebaran Robotics), as shown in 2.1a, is a social semi-humanoid robot that was launched in June 2014. It has the ability to display body language, sense and interact with its surroundings, and move around. Additionally, Pepper can analyze people's expressions and voice tones to recognize human emotions. The robot stands at 1.2 meters tall and moves on three wheels for omnidirectional navigation. It has a total of 20 degrees of freedom (DoFs), including six DoFs in each arm, two each for the head and hip, one in the knees, and three in the base. It also has two RGB cameras, four microphones, and three tactile sensors etc., to perceive the world, as well as two speakers and a touch screen on the chest to convey information to humans. Until July 2018, approximately 10,000 Pepper robots had been sold, with most of its primary application scenarios being in retail, hospitality, and education. Pepper also shows the potential to assist healthcare workers with visitor guidance, patient vital data collection, accompanying patients, and helping some those who are unable to care for themselves in the hospital setting. Studies have already shown that Pepper can also be a good companion for the elderly and help patients with mental illnesses carry out some rehabilitation courses [17]. However, its current use in healthcare is mainly for connecting patients and their families at the reception and allowing doctors to communicate with patients remotely [4] [18]. There is currently no documented use of Pepper in physically assisting patients or the healthcare workers.

The review also referenced a 10-week-long study by Carros et al., which found that older adults who participated in the study enjoyed interacting with the Pepper robot. However, they also emphasized that they did not want robots to replace caregivers. Moreover, understanding the robot's behavior can be challenging for older adults, and they may need to rely on each other to understand the robot's actions. The trust of older adults in robots is also based on the involvement of the care workers. Some technical issues with the robot can also bother, such as long loading time for the robot application and unresponsive touchscreen [19].

NAO

The NAO robot², also developed by SoftBank Robotics, is a smaller humanoid robot, standing at a height of only about 58 centimeters, as depicted in 2.1b. Its latest version is the sixth generation (NAO⁶). It has a total of 25 DoFs, with the DoFs of the upper body being similar to those of Pepper. NAO is equipped with two 2D cameras, seven touch sensors in the head, hands and feet, four directional microphones and speakers for interaction with humans, and the ability to recognize and communicate in 20 different languages. Similar to Pepper, both NAO and Pepper use the open NAOqi platform, which supports multiple programming languages, including Python and C++ [4].

Similar to Pepper, NAO has also been extensively tested and utilized in various scenarios as assistant, such as serving as an teaching assistant for children with autism, a physiotherapeutic assistive trainer for the elderly, a cognitive trainer, and a healthcare assistant [4]. In practice NAO is programmed for a number of teaching and therapeutic behaviors, including singing, exercising, and playing with children. Research has shown that individuals who initially have a fear of robots may face more difficulties, whereas those who are intrigued by robots tend to easily and fluently engage in the same activities [20].

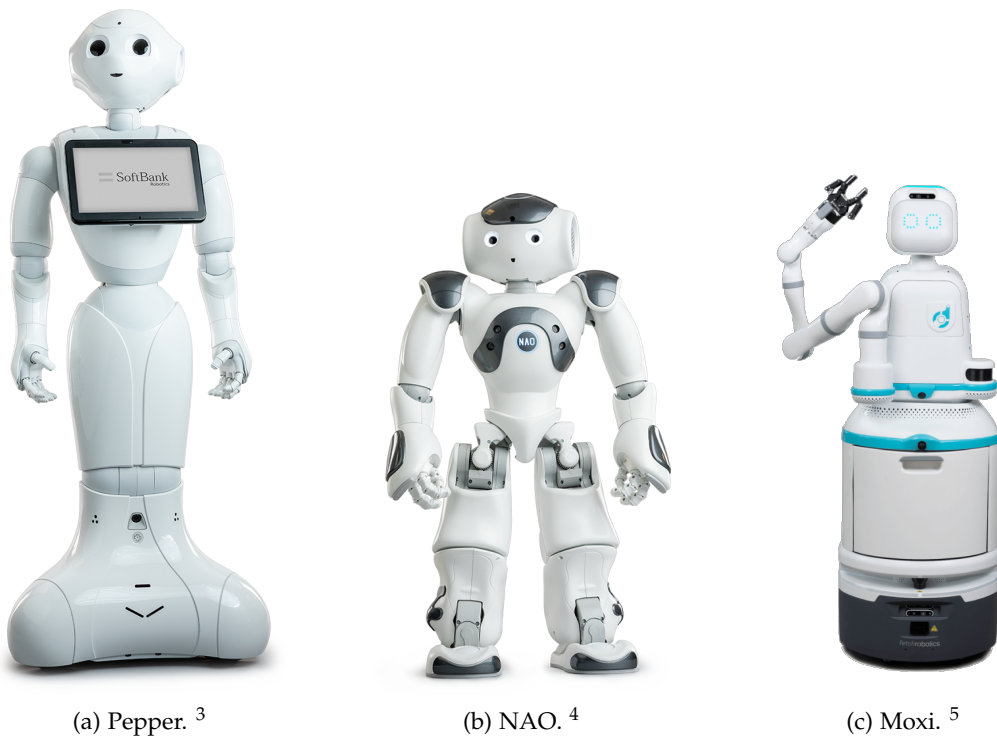


Figure 2.1.: Existing robots in healthcare: Pepper, NAO and Moxi.

²NAO. url: <https://www.aldebaran.com/en/nao>.

³Pepper. <https://us.softbankrobotics.com/pepper>.

⁴NAO. <https://us.softbankrobotics.com/nao>.

⁵Moxi. <https://www.diligentrobots.com/moxi>.

Moxi

Commercial robots such as Moxi, developed by Diligent Robotics, are already being utilized to help with transportation tasks in hospitals. Moxi can assist healthcare workers autonomously with a huge amount of work even outside of working hours. It can retrieve and deliver items to patient rooms and nursing stations, as well as deliver samples to labs and distribute PPE. The robot consists of “a mobile base, a seven-DoF robotic arm with a two-finger gripper, and sensors for environmental perception such as a camera and laser scanner” [4]. Moxi can learn on its own where the rooms and items to be delivered are, and can navigate in the hospital autonomously and safely. It can avoid or give way to obstacles, and can even open elevator doors by itself in order to reach the target location. However, Moxi is only for non-patient-facing work. Its robotic arm is limited to grasping, pulling, opening, and guiding objects [4, 21].

2.1.5. Ethical Concerns

While the use of robots in healthcare can alleviate the burden on healthcare workers and improve the standard of care, it is not without controversy and shortcomings. Ethical issues and social concerns are at the forefront of the controversy surrounding the application of robots in healthcare settings.

In their 2012 work, A. Sharkey and N. Sharkey raised concerns about the ethical implications of using robots in elderly care [22]. While robots can provide companionship, they identified six key ethical issues, including:

1. **The potential reduction in the amount of human contact.** When a robot replaces a human cleaner for tasks such as cleaning floors, it also eliminates the opportunity for social interaction between the elderly and the human cleaner [23]. Extensive research has shown that having a strong social network can help prevent the risk of dementia in the elderly, and frequent social participation, exposure to new experiences, and intellectual stimulation can help maintain cognitive function and reduce the risk of Alzheimer’s disease. Moreover, social contact has been found to reduce stress, which can accelerate the effects of aging. Therefore, reducing contact with humans through the use of robots may have a measurable impact on the health of the elderly, and depriving them of social contact with others through the application of robots may be considered unethical and cruel.
2. **An increase in the feelings of objectification and loss of control.** The primary goal of robot applications in elderly care is usually to improve working conditions for caregivers or reduce costs, rather than prioritizing the well-being of the elderly. When robots replace human caregivers in performing caregiving tasks, elderly may feel a loss of control over their own lives, as the care they receive becomes more robotic and impersonal. This can exacerbate the issue of objectification that is already prevalent in the care of people with dementia. Similarly, elderly who are generally frail and have less autonomy over their lives can be at risk of being treated as inanimate objects when

robots are used to perform caregiving tasks without inherent empathy or care. However, even human caregivers may not always fully respect the dignity of elderly in their care.

3. **A loss of privacy.** Equipped with sensors like cameras and microphones, robots used in elderly care have the capability to easily monitor the activities of the elderly. With current technology, the data collected by these sensors can be easily transmitted to remote locations, and potentially, the entire lives of the elderly can be stored on hard drives. Elderly individuals may not be comfortable with the fact that the operators of these robots could potentially spy on them, even during moments of vulnerability such as when they are disheveled or in the shower. These concerns could be further exacerbated in cases where the mental health of elderly, such as those with Alzheimer's disease, deteriorates, as they may forget that they are being monitored by robots and mistakenly believe they are in a safe and private environment.
4. **A loss of personal liberty.** If a robot is tasked with monitoring the safety of the elderly, it would require a significant level of autonomy to actively intervene and prevent potentially dangerous behaviors, such as leaving the stove on or climbing on furniture to access high cabinets. However, such intervention could prevent seniors from leaving their room or direct them to stay at home at all times. This can be regarded similar to incarceration.
5. **Deception and infantilisation.** Studies argued that any benefits of robot pets or companions are a consequence of deceiving the elderly into mistakenly thinking that the robots are something with which they could have a relationship, and systematically deluding themselves about the real nature of their relation with animals/companions, consciously or subconsciously. However, Sharkey and Sharkey also noted that people may be more accepting of anthropomorphic robots than previously feared [23]. Additionally, there are concerns that interactions with robot toys may lead to the infantilization of elderly people and further disempowerment in dementia care, potentially demeaning their sense of dignity.
6. **The circumstances in which elderly people should be allowed to control robots.** Addressing the aforementioned challenges in using robots in elderly care, a potential solution could be providing older adults with the ability and autonomy to control robots. However, there are complex issues related to responsibility in accidental circumstances, the extent to which the wishes of the elderly should be followed, and the balance between the mental state of the elderly and the level of control of the robot granted to them.

2.2. Use of Simulation in Robotics

2.2.1. Motivation

Experiments in real environments always involve real objects on a physical level. Unfortunately, interacting with real objects always comes with physical risks and potentially higher

costs. This is the problem that need to be faced in the development process of the robots. Simulation in virtual environments provides a practical alternative. In the past decade, as people have become more interested in intelligent robots and the use has greatly increased, Choi et al. argued that virtual environments in computers can be used as experimental grounds to design robots and gain a deeper understanding of the state of robots in a faster, cheaper, and safer way [24].

According to Choi et al., designing a new robot involves two time-consuming stages: the mechanical design and the control policy design. The former is mainly responsible for producing a solution that physically has the ability to complete a set of tasks, while the latter should make the robot “smart” so that it can indeed complete these tasks. Both stages require multiple iterations of the prototype in which the previous version gets improved until the prototype yields acceptable performance. This process can be very expensive, dangerous, and time-consuming. In some cases, it may even be impractical. For instance, designing and testing Mars rovers definitely cannot be done on Mars. Performing iterative loops in a virtual environment can significantly reduce the time spent on these phases [24]. In order to test a new concept or verify an idea, programmers and engineers no longer need to set up collateral systems, add new interfaces or even design new prototypes [25]. For example, in order to test the mobility of a vehicle on different terrains, in reality, the structure of the vehicle needs to be designed repeatedly, and considering the complex connections between the subsystems of the vehicle, this will be tedious and mistake-prone. However, through computer simulation, the modeling and topology of a vehicle composed of complex subsystems can be defined in a template file, and modifying the geometric structure of a vehicle can be quickly achieved simply by altering a few parameters in the template file [26].

Choi et al. also emphasized the safety benefits of simulation in robotics research. Simulation allows for the repeated testing in stress and corner cases. Without risking damage to humans or hardware, researchers can have more freedom in testing. When exploring safety-critical problems in robot development, such as testing a robot’s response to cyber attacks or extreme environments, direct hardware testing can be both costly and potentially unsafe, particularly for large, bulky robots carrying potential technical defects during the development process. The testers could also be exposed to the threats from extreme test environments and the robot itself [24]. In our specific case, as the end-users of the robot will likely be vulnerable patients in a hospital environment with expensive, precision equipment, the risks associated with out-of-control robots are particularly high.

Due to the complexity and cost of manufacturing physical prototypes, it is difficult for most laboratories to acquire multiple devices for shared use. As a result, conducting different experiments simultaneously can be impractical. Employing simulation techniques in virtual environments solves the problem of concurrent use [25]. This could accelerate the development while keeping the budget low and benefit experiments requiring multiple agents.

2.2.2. Limitations

Most simulators typically rely on unrealistic assumptions when conducting simulations. These assumptions include static operating environments, flawless sensing systems, and tasks that do not lead to substantial modifications in the geometry of the operating environment. The most common simulated movement involves simply moving towards a goal point without encountering any collisions. However, in practical scenarios, the tasks performed by real robots are much more complicated. Simulating the interaction between the robot and the operating environment is a dynamic process with significant interdependencies. The robot's actions can fundamentally alter the working environment, and the other way around, changes in the working environment can impact the robot's functionality [27].

The research of Kiesler et al. also implied the potential advantages of using real robots rather than simulated robots for research on human interaction. They observed that participants were more engaged and attentive to their behavior when interacting with a real robot compared to a virtual one [28].

2.3. Digital Twin

In recent years, digital twin (DT) has gained significant attention from researchers, as evidenced by the exponential growth in publications featuring DT as a keyword since 2016. This surge in interest can be attributed in part to Gartner listing DT as one of the top 10 strategic technologies for three consecutive years from 2017 to 2019 [29]. With 20 billion connected sensors and endpoints estimated in 2019 by Gartner for 2020, there may be billions of items with their own DT ⁶.

2.3.1. Definitions and Concepts

While DT technology has just gained popularity in recent years, the concept itself is not new. NASA⁷'s simulation of extreme conditions for the spacecraft and its components during the Apollo 13 mission in 1970s is often considered as a predecessor to the DT model. A similar concept known as the "mirror world" was imagined as early as in 1991. However, due to technical limitations such as low computing power, limited connectivity, undeveloped machine algorithms, and data storage and management challenges, DT did not find practical applications in earlier times [29, 30].

The concept of DT was first introduced by Michael Grieves in 2003 at the University of Michigan Executive Course on Product Lifecycle Management (PLM), but the term "digital twin" was still not clearly defined at that time [31, 32]. The first definition of DT was forged by NASA in 2010, although it was specific to vehicles and aerospace [29]. In 2014, Grieves

⁶Gartner Identifies the Top 10 Strategic Technology Trends for 2019. <https://www.gartner.com/en/newsroom/press-releases/2018-10-15-gartner-identifies-the-top-10-strategic-technology-trends-for-2019>.

⁷National Aeronautics and Space Administration. It is a United States government agency that is responsible for science and technology related to air and space. <https://www.nasa.gov>.

formalized the DT model in his white paper, defining it as consisting of three main elements: a) physical products in real space, b) virtual products in virtual space, and c) connections of data and information that tie the virtual and real products together [31]. Conceptually, a DT simulates the state of its physical twin in real-time and vice versa. However, due to the lack of standardization, DT has multiple definitions and is often confused with other related terms such as digital model or software analogue in literatures [29].

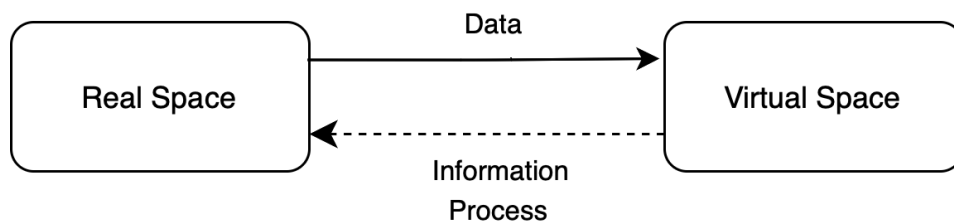


Figure 2.2.: Model of a digital twin. This picture is adapted from Grieves' model. [31]

In order to reduce the confusion around the term, Singh et al. provided a universal definition for DT [29]:

“A Digital Twin is a dynamic and self-evolving digital/virtual model or simulation of a real-life subject or object (part, machine, process, human, etc.) representing the exact state of its physical twin at any given point of time via exchanging the real-time data as well as keeping the historical data. It is not just the Digital Twin which mimics its physical twin but any changes in the Digital Twin are mimicked by the physical twin too.”

2.3.2. Advantages

DT technology has gained significant attention due to its numerous advantages. As per the summary provided by Singh et al. [29], some of the reported advantages of DT include:

1. **Speed prototyping and product redesign:** Simulation in a virtual environment with DT allows for investigation of multiple scenarios, shortening the design and analysis cycle of a product and making prototyping or redesign faster and easier. As the DT remains connected to the physical twin throughout its lifetime, engineers and product designers can compare actual and predicted performance at any time, enabling them to reconsider and improve the products they design [33].
2. **Cost-effective:** DT can reduce the overall cost of prototyping over time. Traditional prototyping involves physical materials and labor, and redesigning a product can be time-consuming and expensive. Destructive testings can be especially expensive because

it destroys the physical prototype. In contrast, DT allows for testing products under different scenarios, including destructive scenarios, without any additional material cost, thus avoiding material waste and reducing cost. As a result, DT can reduce operating costs so that the lifetime of equipment and assets can be extended [31].

3. **Predicting problems and system planning:** DT enables prediction of the future state of its physical twin and potential problems and errors, allowing for proactive system planning. Real-time data flow between the physical twin and the DT enables prediction of problems at different stages of the product life cycle. This is particularly beneficial for products with multiple components, complex structures, and multiple materials, which can be challenging to manage using traditional methods [34]. Traditional maintenance methods are often based on heuristic experiences and worst-case scenarios, and are reactive rather than proactive [35]. However, DT can anticipate defects and damage to products/systems, allowing for scheduled maintenance in advance. By simulating different scenarios, DT can provide the best solution or maintenance strategy, making product/system maintenance easier.
4. **Accessibility:** DT allow users to remotely control and monitor their physical twins. Virtual systems, such as DT, unlike physical systems, can be widely shared and accessed remotely [31], without being restricted by geographic location. Remote monitoring and control of systems becomes particularly valuable in situations where local access is limited, such as during the COVID-19 pandemic, where public health policies implemented by many governments necessitate remote or contactless work as the only viable option [36].
5. **Safer than its physical twin:** In industries with extreme or dangerous working conditions, such as oil wells, mines, and other hazardous environments, the ability to remotely access physical twins via DT and its predictive nature can reduce the risk of accidents and dangerous failures. Remote monitoring without human contact can also help keep healthcare workers and patients safe during the COVID-19 pandemic. According to a 2020 Gartner survey, nearly one-third of companies are using DT remotely to improve employees and customers safety during the pandemic ⁸.
6. **Training:** DT can be used to develop more effective and safer training programs compared to traditional methods [37]. Operators can use DT for training before working on high-risk sites or dangerous machines, which helps them deal with the same situations in person as they can be exposed to different processes or hazard scenarios in a virtual environment. DT can also be an excellent tool for bridging the knowledge gap from experienced workers to newcomers.

⁸Gartner Survey Reveals 47% of Organizations Will Increase Investments in IoT Despite the Impact of COVID-19. <https://www.gartner.com/en/newsroom/press-releases/2020-10-29-gartner-survey-reveals-47-percent-of-organizations-will-increase-investments-in-iot-despite-the-impact-of-covid-19->

2.3.3. Classification

DT can be classified into different types based on various criteria. Grieves and Vickers propose a classification based on the creation time within the product life cycle, resulting in two types [38]:

1. **Digital Twin Prototype (DTP):** DTP is created during the design phase prior to the physical prototype. The product cycle begins with the development of a DTP, which undergoes various tests, including destructive ones, before the physical twin is built. DTP helps identify and prevent unforeseen and undesirable situations that may be challenging to detect with traditional prototypes. Once a DTP is validated, its physical twin can be fabricated in the real world. The accuracy of the simulation/model determines the quality of the physical twin.
2. **Digital Twin Instance (DTI):** DTI is created during the production phase after the product is ready. This type of DT remains connected to its physical counterpart throughout its lifespan. DTI enables the exchange of data between the real and virtual spaces for monitoring and predicting system behavior. This data allows for the assessment of whether the system exhibits the expected behavior and whether potential issues have been successfully mitigated. The bidirectional link between the two systems ensures that any changes made in one system are replicated in the other.

Kritzinger et al. propose a classification based on the level of integration [39]:

1. **Digital Model:** In this type of DT, data exchange between physical and digital objects occurs manually. Changes in the state of physical objects are not directly reflected in their digital counterparts, and vice versa.
2. **Digital Shadow:** In this type of DT, data flows automatically from physical objects to digital objects, but data flow from the digital side to the physical side requires manual operation. Consequently, changes in the state of a physical object are directly reflected in its digital counterpart, but not the other way around.
3. **Digital Twin:** This type of DT enables bidirectional automatic data exchange between physical objects and digital objects. Changes in either the physical or digital object trigger corresponding changes in the other.

DT can also be classified based on its application focus [29, 38]:

1. **Predictive DT:** Used for predicting future behavior and performance of the physical counterpart.
2. **Interrogative DT:** Used for examining the current or past state of the physical counterpart.

Furthermore, DT can be classified based on if the focus of application is on product, process, or performance⁹. According to the magnitude involved in manufacturing, DT can also be classified from a hierarchal perspective, into different levels [40].

2.3.4. Challenges

Novelty of the Technology

DT technology is still in its early stages and faces complexities in terms of popularization and implementation. While it holds great potential, its value for individuals, enterprises, and industries is not yet fully understood, and the absence of successful practices and business models makes it challenging to estimate implementation costs. Related technologies like 3D simulation, Internet of Things (IoT), AI, and big data, which are closely associated with DT, are still in development themselves, hindering the progress of DT. Real-time virtual-real interaction technology and intelligent perception and connection technology are also crucial for DT implementation and are yet to be developed.

High Costs

The development of DTs requires ultra-high-fidelity computer models, which makes it a time-consuming and labor-intensive exercise. Furthermore, their high demand of computing power also contributes to the cost of DT, making DTs an expensive investment. Embedding existing systems into data-collecting sensors, along with the need for large-capacity storage devices and high-performance hardware and software, adds additional expenses beyond the DT itself. West and Blackburn estimated that completely implementing digital threads/digital twins of weapon systems for the US Air Force would likely cost trillions of dollars and take hundreds of years, making it impractical to achieve the technology on such a scale.

Absence of Standards and Regulations

Due to the novelty of the technology and the various definitions found in literature, there is a lack of sharing and mutual understanding of interfaces and efficient design of data flow in DT technology. Standardized models, interfaces, protocols, and data are crucial for efficient collaboration, particularly in industries such as aerospace, automotive, and healthcare. Additionally, the technologies on which DT relies, such as big data and artificial intelligence, are still in their early stages, and the corresponding laws and regulations are not yet fully developed. While a standardized framework for DTs, ISO 23247 (Digital Twin Manufacturing Framework), has been introduced in 2021, including a Reference Architecture (RA) for developing and implementing digital twins in manufacturing, challenges remain in implementing the RA, and some applications are still in their early stages. The current reference architecture lacks support for two essential functions: data storage and digital twins [41].

⁹Digital Twin. <https://www.plm.automation.siemens.com/global/en/our-story/glossary/digital-twin/24465>

Data-related issues such as privacy, confidentiality, transparency, and ownership also raise concerns due to the extensive data processing involved in DTs. Additionally, there may be discrepancies in the lifetimes of DTs and their physical counterparts.

2.4. Interaction with Smartphone

2.4.1. Remote Interaction

A massive amount of research has been conducted on remote interactions, particularly involving interactions with large screens located at a distance. These works typically include two primary components: pointing technologies and interaction technologies. The general approach in much of these works involves selecting specific objects on a distant screen using pointing technologies and subsequently engaging with these objects through various interaction techniques. Notably, the interaction techniques involving smartphones have served as an inspiration for our work.

Direct Pointing (Lazer Pointer)

Laser pointer-based pointing has been found in various studies due to its cost-effectiveness, accessibility, reliability, and intuitiveness. Olsen and Nielsen examined direct pointing with a laser pointer for interaction with virtual objects. Users either held a laser pointer or attached it to the back of a glove to direct it towards objects on a projected screen. Cameras were employed to detect the laser pointer's position, allowing users to interact with objects by keeping the laser pointer above them or by deactivating the laser pointer [42].

Seifert et al. proposed a more versatile solution named PointerPhone, also based on a laser pointer. They designed a new mobile phone prototype with the laser pointer attached to the phone's front end. Similar to Olsen and Nielsen's approach, users point to objects for interaction, and an exo-centric camera detects the laser pointer's position on the screen, and then access the selected object. With the assistance of smartphone hardware, users could have various interaction options, such as tapping, sliding, or tilting the phone [43].

While the methods implemented by Olsen and Nielsen and Seifert et al. allowed for precise interaction with objects on remote screens, this mainly relied on the laser pointer's direct pointing and the external camera, with limited involvement of the smartphone itself. This solution is not practical for our development since we cannot rely on externally fixed cameras. Additionally, we lack a fixed screen for laser pointer to point at. Nonetheless, their work served as inspiration, and in our final solution, we utilized the smartphone as an implication for ray casting pointers.

Gesture-Based

With advancements in computer vision algorithms, gesture-based pointing methods have gained more attention. Bragdon et al. presented a remote screen interaction approach similar

to the work of Seifert et al. Users can point to content on the screen and interact with them using their smartphones and specific gestures. This approach, like the previously introduced examples, relies on exo-centric depth cameras for recognizing users' gestures [44].

Gaze

Gaze-based remote interaction has been an area of early interest. Jacob proposed a technology that allows interaction through eye movements. An impressive aspect of this interaction method is its ability to respond to a user's intentions even without explicit commands. This makes it effective when users cannot use other means to input commands. However, Jacob suggested that relying solely on eye movements might be more suitable for target selection, while executing commands is better confirmed through buttons rather than gaze on an object with long dwell time [45].

Sibert and Jacob conducted experiments comparing gaze with dwell-time interaction to mouse-based interaction and found that gaze with dwell-time was faster [46]. Zhai et al. introduced a gaze-based interaction method called MAGIC, which combines traditional gaze with manual control, using gaze to approximate the desired position of the cursor and then use regular manual input devices for selecting and clicking the target. According to their findings, MAGIC reduces physical effort and fatigue compared to traditional manual pointing and offers faster interaction speed [47].

In our implementation, we have adopted gaze as the method for button selection in AR HMD. However, as we use AR HMD that can track the user's head orientation, we have simplified gaze by using the user's forward-facing head orientation as the gaze direction.

Inertial Sensing

With the development of Micro-Electro-Mechanical Systems (MEMS), an increasing number of devices are equipped with various motion sensors. The Wii Remote, as a well-known commercial remote controller, utilized MEMS accelerometers and infrared sensors to calculate the controller's position and orientation relative to the screen, enabling interactions with game content¹⁰.

While specialized controllers are powerful, they are not always carried by users and may not be suitable for interacting in various settings [43]. Most modern smartphones are equipped with inertial measurement units (IMUs), making them powerful alternatives to specialized controllers. There have been numerous attempts to use the orientation and movement of smartphones for remote screen interactions, which will be discussed in detail in the following chapter 3.

¹⁰Video game system with wireless modular handheld controller. <https://patents.google.com/patent/US8313379B2/en#patentCitations>

2.4.2. 3D Interaction

In addition to touchscreens, most modern smartphones are equipped with various sensors such as accelerometers, gyroscopes, and cameras. Using these sensors for 3D interactions with smartphones is an area of active research. In this section, we will discuss various methods of using smartphones for 3D interactions in detail.

Smartphone Tilting and Pointing (3 DoF)

Smartphones are equipped with accelerometers and magnetometers, which can effectively detect the direction of Earth's gravity, allowing for the calculation of the phone's current orientation. The gyroscope in smartphones can measure changes in orientation by detecting angular acceleration. Gyroscope data, particularly the high-quality data provided by iOS devices, are accurate and stable, requiring minimal post-processing [48].

Vajk et al. attempted to use smartphones with 3D motion sensors as "Wii-like" controllers for playing games on large public displays. They successfully used the smartphone's accelerometer to control games by tilting the phone. However, they noted that although this interaction method was novel and engaging, the motion sensing capabilities of smartphones were not yet comparable to the Wii Remote [49].

Katzakis and Hori's work also explored using smartphones as 3-DOF controllers. Users could rotate the phone to rotate the virtual target object, and this control method was found to be easier to learn compared to using a mouse or stylus. They used data from sensors such as accelerometers and magnetometers [50].

Graf and Jung utilized gyroscope data to create a virtual laser pointer. After calibrating the phone's orientation and position, users could change the direction of the pointer by tilting the device. Rotating the device along the axis of the laser pointer allowed control of the cursor's depth in 3D visualization [48].

Our development builds upon these works, utilizing the reliable tilt detection of smartphones to provide a novel control scheme for Robody's arms, which will be discussed in detail in the following chapter.

Gaze and Touch

Smartphones equipped with touch screens offer more interaction possibilities. Gaze and touch technology involves users gazing at a target on the screen and then using the touchscreen for precise cursor positioning or interaction with the selected object. Gaze and touch is a technique that combines the features of direct touch and indirect touch. Users can perform immediate actions when touching without the need to move the cursor beforehand.

Stellmach et al. proposed an interaction technique that combines gaze and touch input from handheld devices. This technique also employs the previously mentioned MAGIC method, where users use gaze to determine the rough cursor position and then fine-tune the cursor's position or confirm selection using the touchscreen of the handheld device [51].

Pfeuffer et al. discussed the cost and efficiency of gaze and touch. Through experiments, they found that gaze and touch, compared to pure touch, were slower and less accurate in

tasks involving dragging objects. However, they were equally fast in tasks involving rotation and scaling objects, where gaze and touch were even more accurate [52].

Existing studies have focused on exploring the efficiency and accuracy of gaze and touch as an interaction method. However, our work is more concerned with the possibilities this interaction method offers in replacing specialized controllers with smartphones. Our development draws inspiration from this interaction method, where gaze is used to select targets, followed by pressing a confirmation button on the smartphone screen.

Smartphone 6 DoF Tracking

The IMUs typically found on smartphones are usually only suitable for discrete motion detection and are challenging to provide precise data necessary for 6 DoF tracking. Any measurement errors, no matter how small, accumulate over time, resulting in "drift," which is the increasing difference between the device's actual position and where it's considered to be located [53]. However, smartphones equipped with cameras can offer new possibilities.

Rekimoto proposed a method that uses printed 2D matrix markers (square barcodes on paper) attached to objects in the environment to determine the smartphone's position and orientation. Using a smartphone application, the camera can locate and identify external markers to estimate the camera's relative position and orientation [54].

Modern smartphones, with their more powerful cameras, can use advanced computer vision algorithms (such as dense SLAM [55]) to detect the precise surface geometry of an unknown environment and use this data to estimate their own position and rotation. With the help of ARKit (Apple's AR toolkit, introduced in detail in subsection 3.2.2), Babic et al. developed a smartphone application called Pocket6. In this application, ARKit, using the data produced by IMU sensors and images from the smartphone camera, calculates the device's spatial position and rotation, converting physical movements into movements within the virtual environment [53].

Babic et al.'s work had a profound impact on our work. Similarly based on ARKit, we followed their approach to enable 6 DoF motion tracking on smartphones and developed a Robody hand control method based on this foundation.

2.5. Human-Robot Interaction (HRI)

The core of the thesis lies in exploring human-robot interaction (HRI) interaction modes through touch screens and augmented reality (AR) devices to achieve a more intuitive, user-friendly, and efficient operation. As a result, various HRI modes along with their pros and cons will significantly impact our work. Specifically, interaction modes utilizing AR and touch screens (smartphones) hold crucial guiding principles and reference value for our study and development.

2.5.1. HRI Styles

Over the past few decades, the field of HRI has dedicated significant efforts to investigating the dynamics of human-robot interaction and exploring effective means of interaction between humans and robots. Especially as the utilization of the home robots continues to expand, numerous companies and laboratories are actively seeking innovative approaches which enable human to control the robots efficiently and easily, indoor and outdoor [56].

Phaijit et al. extended the HRI styles proposed by Rekimoto and Nagao [57] and proposed an taxonomy of HRI styles that include augmented HRI and encompass bidirectional interactions among user, real-world, and robot entities [58], as shown in Figure 2.3.

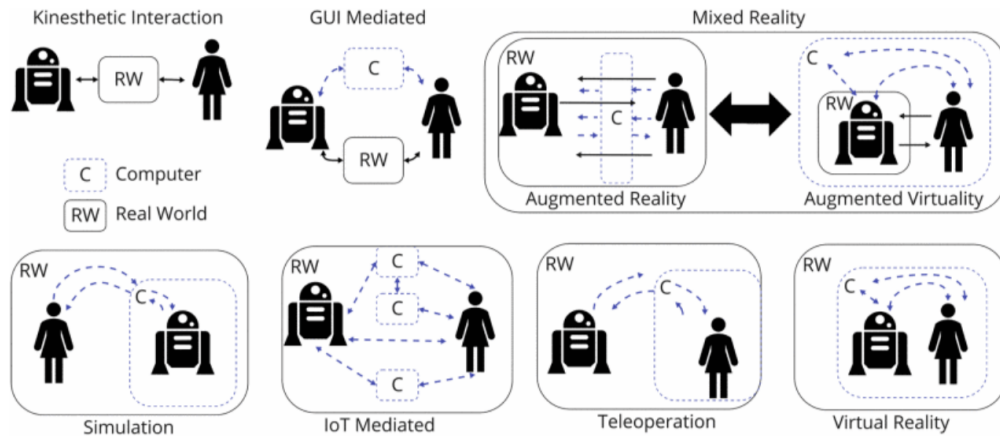


Figure 2.3.: Different HRI styles extended from [57] by Phaijit et al.[58].

According to Phaijit et al. [58] and Rekimoto and Nagao [57], HRI styles can be divided into:

1. **Kinesthetic Interaction:** Kinetic Interaction is a fundamental human-to-human or human-to-robot interaction that involves direct physical engagement through force and movement. It occurs without the involvement of computerized mediations [58]. Examples of Kinetic Interaction include teaching of manual skills, collaborative lifting and moving of heavy objects [59], and the use of robots to provide haptic feedback during interactions [60].
2. **Graphical User Interface (GUI):** The vast majority of robotics platforms have a GUI as a medium between human and robots. These GUIs provide users with predefined commands or authoring tools which enables the users to communicate and interact with the robots without having in-depth knowledge of the technical or programming details underlying the operations. GUIs are often deployed on tablets to facilitate the interaction and compensate for limitations in natural language processing (NLP) capabilities which are not robust (e.g. Pepper, as mentioned before, shown in 2.1a). While GUIs can be powerful tools, they introduce an interaction gap between the user

and the robot, as the interaction is always mediated by the computer, and there is no direct interaction between the two parties [58].

3. **Teleoperation:** The teleoperation of robots is a prominent approach in HRI. Extensive research has been conducted to explore different types of controllers that enhance teleoperation, aiming to improve efficiency and provide a more natural experience for the operator. These controllers include joysticks, gamepads [61], haptic feedback devices [62], and others [58].
4. **Internet of Things (IoT) Mediated:** IoT technologies, including ubiquitous sensors, enables robots and users to establish real-world interactions through multiple connected channels [58]. Discussed in a survey conducted by Simoens et al., combining IoT technologies with robotic systems has significant potential and advantages [63].
5. **Simulation:** As previously discussed, simulation plays a crucial role in robotics. Phaijit et al. emphasized that there are very limited direct interactions between robots and the real-world, and most systems rely on predetermined parameters to generate simulated robot behaviors [58]. A notable example of such a simulation system is USARSim, which enables the interaction between humans and virtual robots [64].
6. **Virtual Reality (VR):** Interactive virtual environments host both robot agents and the user to collaborate on a task, offering immersive virtual space for the human-robot interaction. A key aspect of these environments is the establishment of a sense of presence, allowing users to feel fully immersed and coexist with virtual robots. Unlike AR settings, immersive VR environments limit the interaction with real-world entities, focusing on creating a self-contained and highly interactive virtual space [58].
7. **Mixed Reality (MR):** MR technology brings together digital and real-world content, and its coupling with robotic entities offers various possibilities for HRI. In recent years researchers and developers have explored diverse scenarios and applications in the field of HRI [58, 65].

Within these categories, GUI, Teleoperation, Simulation, and AR are the styles that hold our attention. Acquiring a comprehensive understanding of the characteristics of these styles aided us in seamlessly integrating them into a unified platform in the most optimal manner, which is the focal point of our work.

2.5.2. Challenges in Robot Teleoperation

Despite significant advancements in autonomous robotics, the current technological landscape does not support complete autonomy for robots in complex human-robot interaction scenarios. Adamides et al. recognized that certain domains, such as medicine and agriculture, involve highly unstructured tasks that still heavily rely on human knowledge and experience. In these contexts, autonomous robots are not yet capable of adjusting their acts to specific requirements.

As an alternative, teleoperated robots exploit human experience while protecting the operator in hazardous environments [66].

In our case, Robody encounters challenges when interacting with patients and complex clinical environments. This complexity introduces risks and unpredictability if Robody is granted excessive autonomy without ensuring safety. The potential contagiousness of diseases carried by patients can also endanger healthcare workers. As a result, we need effective remote control methods for such situations.

However, teleoperating a robot can be a challenging task for novices, as it often demands a high level of skill and attention from the teleoperator [67], and many other human performance issues have been identified when teleoperating robots.

Limited Field of View (FoV)

One major issue is that traditional cameras have a limited field of view (FoV), which can negatively affect the operator's perception of the environment in multiple ways [68]. In a study comparing driving an armored vehicle with head-mounted displays (HMD) versus periscopes, it was observed that an HMD system which has wider FoV improves vehicle control and leads to faster mission completion times [69].

Pose and Depth Estimation

Another challenge is understanding the teleoperated robot's orientation and pose. Due to the lack of a perception of gravity and familiar reference objects, it can be challenging for the operator to accurately estimate the pose of the robot [68]. Furthermore, the perception of depth information can be degraded, which affects the teleoperator's estimation of distance and object size [70]. Monocular cameras can only project 3D depth information onto a 2D display surface, resulting in a compression of depth perception [71]. This effect is especially pronounced for robots that move on the ground because of their low viewpoint [68]. Studies have shown that humans underestimate distances more in virtual environments than in the real world [72]. With a monocular camera, the teleoperator must rely on interposition, light and shadow, linear perspective, and other cues to judge the depth of the remote scene [73], which can be especially challenging in unfamiliar or cluttered environments [74].

Low Video Quality

Limited bandwidth for remote video streaming can result in a low frame rate and poor video quality, resulting in difficulties in predicting the motion of the robot [68].

Multi-DoF Robot Control

Humanoid robots, designed to resemble the physical structure of humans, possess a multitude of DoFs. The incorporation of multiple DoFs is common in the development of robots capable of picking up and deliver objects [75]. Multiple DoFs provide robots with the ability to move and rotate along various axes (x , y , and z) or any combination of these axes within the

three-dimensional space. For instance, robotic arms commonly feature 4 or 6 DoFs. As the number of DoFs increases, the complexity of motion and the associated control challenges also intensify, posing difficulties even for experienced operators [76].

To ensure accessibility to a wider range of users, the teleoperating system should offer an intuitive interface for simultaneously controlling multiple DoFs of the robot. Furthermore, the system should not depend on external sensors or controllers to be adaptable to various environments [77].

At present, prevailing methods for controlling a multi-DoF robot involve the utilization of a joystick or gamepad [61, 78]. Nonetheless, these control strategies have inherent limitations, as the number of controllable DoFs is constrained by the available human input on the control device. The mismatch between the number of inputs and DoFs further adds complexity to the design of the control algorithm. Although employing a combination of two or three inputs can increase the controllable DoFs, it also compromises operability and necessitates more extensive user training [76, 79].

In addition to conventional control methods, researchers have also explored alternative approaches for controlling humanoid robots, such as the use of control suits [80] and VR technology [77, 81], which will be discussed in the following part. It's worth noting that currently, there hasn't been an existing robotic control mode similar to ours, which utilizes both touchscreen and AR technology as tools.

2.5.3. Virtual Reality (VR) in HRI

As previously discussed in the context of simulation, there are numerous scenarios involving the interaction between users and simulated robots. These situations include manufacturing training exercises, remote manipulation interfaces, or design iterations during the development process. Studies in these applications are often run on screens. Nevertheless, the utilization of VR for interacting with simulated robots presents an appealing alternative. VR offers enhanced visual cues of real environments, thereby providing a more immersive and realistic experience for users [82].

Tang and Yamada developed a robotic system for construction robots using virtual reality. Their experiments confirmed that their design is superior in operability, safety, and reduction of stress compared to traditional 3-screen displays [83].

Pausch et al. conducted pioneering research aimed at quantifying the advantages of VR over traditional displays in terms of immersion [84, 82]. Through experimental investigation, they successfully demonstrated that users utilizing VR were able to establish a mental frame-of-reference for the space more rapidly during a search task in heavily camouflaged scenes. Kulshreshth and LaViola focused on exploring the player performance benefits associated with head tracking in modern video games. Their study revealed notable performance advantages for expert gamers, particularly in shooting games, as well as enhanced enjoyment in slower-paced video games such as flight simulation games [85].

The experiment of Liu et al. also compared VR displays to on-screen displays, and assessed the effects of head tracking and stereoscopic perception on user performance. Their findings demonstrated the significant influence of VR displays on users' perception of robots. VR

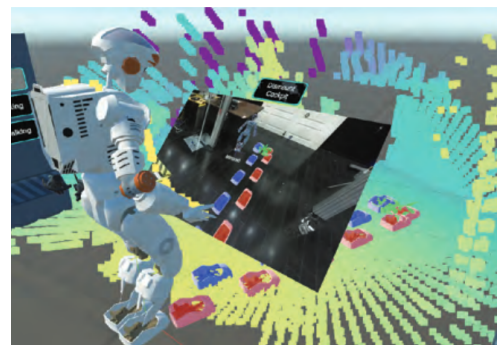
technology offers users enhanced depth cues and stereoscopic visual cues [82].

The VR cockpit interface (shown in 2.4a) proposed by Jorgensen et al. for controlling NASA's humanoid robot Valkyrie serves as a significant reference for our work. They addressed the inefficiency of the human-computer interface for explosive ordnance disposal (EOD) tasks using a mouse and keyboard, with approximately half the time being spent on waiting for the operator's commands [86]. Their development focused on enhancing the efficiency of robot control and introduced a novel VR interface comprising cockpit mode and a floating AR screen fixed with respect to the robot (shown in 2.4b). These modes allow users to "ride the skin" or "zoom out" of the robot, providing improved situational awareness and seamless multimodal sensor data integration in a mixed reality environment [81].

The interface offers various control modes, including locomotion mode, manipulation mode, and high-level commands mode. Locomotion mode and high-level commands mode enable interaction through virtual panels, while the relatively unsafe whole-body IK streaming requires strict voice command recognition or external robot operating system (ROS) command. When controlling the robot's locomotion, users can choose to project a waypoint on the ground, and then the user can preview the robot's planned footsteps and make decisions accordingly. Users can also manually cast a left or right footstep using trigger buttons or just use the joystick to control the robot's walking. Whole-body IK tracking can be enabled or disabled using voice commands such as "thaw" or "freeze," and partial-body tracking is also possible with more detailed voice commands. Their experiments demonstrated that an HMD and two controllers are sufficient for most upper-body humanoid manipulations. They employed the relative transformation between the initial tracker pose and the current tracker pose to calculate the robot's IK target, which provides a safer and more user-friendly incremental adjustment ensuring a comfortable workspace. High-level commands, such as placing the robot to default positions or changing the configurations, as well as accessing stored poses like hand power grasp, are executed using controller buttons and voice commands [81].



(a) VR cockpit interface [81].



(b) UI with a floating AR screen [81].

Figure 2.4.: VR interface developed by Jorgensen et al. [81].

2.5.4. Mixed Reality (MR)/Augmented Reality (AR) in HRI

One of the most popular definitions of mixed reality (MR) is originated from the virtuality continuum proposed by Milgram and Kishino in 1994. The continuum illustrates a spectrum between the real world and the virtual world, and when the two are blended together, anywhere along the reality-virtual continuum, including AR, it is referred to as mixed reality [87].

Similar to VR, AR can either propose to model the real world based on an environment that imitates or symbolizes the real world, or it can create artificial environments that do not correspond to anything that exists. Based on this distinction, Hughes et al. proposed a functional taxonomy of AR environments. They divided AR into two categories based on functionality: those that focus on improving our understanding of the real environment, and those that just add a virtual environment that isn't based on reality. The former visualizes objects, existences or relationships that exist in reality but cannot or are difficult for users to perceive; the latter can generate a virtual environment without being restricted by reality [88].

Green et al. conducted an experimental study comparing three user interface techniques for teleoperating robots. They developed a multimodal system that allowed users to perform robot control tasks with different interfaces. The three user interfaces investigated included: (1) Typical teleoperation mode with a single ego-centric camera feed from the robot; (2) limited AR-HRC interface that allows users to see the robot in the work environment through AR; (3) full AR-HRC interface that allows users to see the robot in the work environment through AR, to work with the robot with voice and gestures, and enables plan creation and review prior to execution. A typical user interface for teleoperating a robot is the first one mentioned above, i.e. using visual cues from the camera. However, the authors argued that this ego-centric view often makes it challenging for operators to maintain awareness of the robot in its surroundings. Their experiments demonstrated that, objectively, the AR-HRC interface achieved higher accuracy, and the completion time using the direct teleoperation interface was shorter. Subjectively, users reported a greater sense of collaboration when using the AR-HRC interface, considering the robot as more of a partner in the human-robot collaboration. Users also experienced improved perceptual awareness, which aligned with the higher accuracy. The majority of users rated the AR-HRC interface as the most effective among the three interfaces tested [89].

Head-mounted displays (HMDs), such as the Microsoft HoloLens [90], are a type of device commonly used to experience AR. According to Oh et al., these HMDs possess several key characteristics: (1) they enable wearers to perceive virtual objects while simultaneously engaging in real-world tasks; (2) they ensure that the displayed virtual objects are visible exclusively to the wearer; (3) HMDs are designed to be utilized in mobile environments,. However, Oh et al. also highlighted that, despite these capabilities, the ideal form of interaction for exploiting the features of AR HMDs remains unclear [91].

Cognitive Load

Several studies have examined the impact of AR technology on the cognitive load of operators [92]. Suzuki et al. asserted that AR can provide visual feedback within a person's line of sight, tightly coupled with the physical interaction space. This coupling reduces the user's cognitive load when shifting attention and context between the robot and an external display. They further highlighted the increasing interest in integrating AR into robotics over the past decades to enhance their inherent visual and physical output capabilities [65].

The work of Tang et al. also demonstrated that AR systems can improve task performance and alleviate the mental workload of assembly tasks. However, they also acknowledge that current calibration techniques and tracking technologies present significant limitations for practical AR implementation. Users may struggle to manage their attention effectively within such systems, potentially over-relying on cues provided by AR. If the abundance of AR cues overwhelms the user's attention, it may lead to the phenomenon of attention tunneling, diverting attention from crucial information in the physical environment and impairing task performance [93].

Hardware Limitations

However, despite the advantages of AR-based controls, there are several challenges that need to be addressed. Modern commercial AR HMD devices typically incorporate gesture recognition and hand tracking capabilities [90, 94]. A user study conducted by Tung et al. revealed that individuals wearing AR HMDs often prefer non-touch and non-handheld interaction such as in-air-gesture controls [95]. However, gesture recognition and hand tracking through the built-in cameras of AR HMDs usually require users to raise their hands above their face, which can lead to arm fatigue when maintained for extended time [91]. Additionally, Cardenas et al. proposed that continuous gesture-based interactions may be impractical or exhausting for astronauts wearing heavy spacesuit gloves [92]. Furthermore, concerns regarding social acceptance have led users to prefer less conspicuous interactions, such as gesture interactions performed in front of the torso [95].

Cardenas et al. also discussed the limitations of the AR hardware they employed, namely the Magic Leap 1, which coincidentally is the same device utilized in our own work. The Magic Leap 1 specifications impose certain constraints on the FoV available to users, with horizontal, vertical, and diagonal FoV of 40, 30, and 50 degrees respectively. Furthermore, the software SDK set the near clipping plane at a distance of 37 cm. Consequently, objects that are in close proximity to the user will vanish from view. As a result, gestures or interactions with 3D virtual objects occurring outside the FoV may not be adequately captured, introducing limitations on the user's level of immersion. [92].

2.5.5. Touchscreen/Mobile Phone/Smartphone in HRI

Touchscreen interface is a combination of a display device and an input device [96]. Touchscreen interface was first introduced more than 50 years ago [97]. Since then, touchscreens

have gained widespread usage and have become an integral part of various industries, commercial settings, and consumer applications. These applications include industrial controls, medical equipment, home appliances, and smartphones [96].

Touchscreens are undeniably appealing due to their ability to integrate display and input functionalities, resulting in space and design efficiencies. They also offer human factors and ergonomics (HFE) advantages by enabling direct mapping of inputs to the intended targets [96]. However, the use of touchscreens introduces certain HFE compromises, including the absence of haptic feedback, and ergonomic compromises such as user fatigue and discomfort [98, 99, 100]. Additionally, the accuracy of input-to-target mapping is susceptible to issues such as the "fat finger problem" [101]. The absence of haptic feedback might be partially compensated by vibration motors, but it still doesn't match the feeling provided by actual physical buttons or joysticks.

Over the past two decades, smartphones have experienced tremendous growth in popularity and functionality. Mobile devices have evolved from simple communication tools to small pocket personal computers operating systems, equipped with touch interfaces and voice-command capabilities. They not only utilize the GSM/UMTS cellular phone networks but also Bluetooth and Wi-Fi to exchange data with other electronic devices [102]. Smartphones offer a distinct advantage as ubiquitous devices for interacting with robots due to their proper computing power and the usability of multimedia [103].

Numerous attempts and applications have been made to control robots using mobile phones or smartphones. Before the maturity of smartphone technology, studies on robot control through mobile phones were already underway. As early as 2003, Sekmen et al. successfully developed a HRI mechanism that allowed human commanders to control robots using mobile phones (shown in 2.5a). The system offered two primary modes: manual control and automatic control. In manual control mode, users had complete control over the robot, while in autonomous control mode, users could activate specific built-in functions of the robot. However, the small screen size of mobile phones at that time posed a challenge, and the authors recognized the importance of efficient GUI design that could maximize the use of the limited screen [104].

Cho et al. conducted research utilizing the physical keypad of mobile phone available at the time to directly control robot movement (shown in 2.5b). By connecting the user's mobile phone to the mobile robot via a voice call, the robot moves corresponds to the buttons pressed by the user [56].

With the development of touchscreen smartphones, the focus shifted to using these devices for manipulating robots. Researchers [105, 103, 106] followed a similar design approach to the work of Cho et al. (shown in 2.5c). They employed virtual buttons on smartphone interfaces to enable real-time control of robot movement. Furthermore, they leveraged the advanced display capabilities of smartphones to transmit and display real-time video information captured by the robot's camera on the smartphone interface.

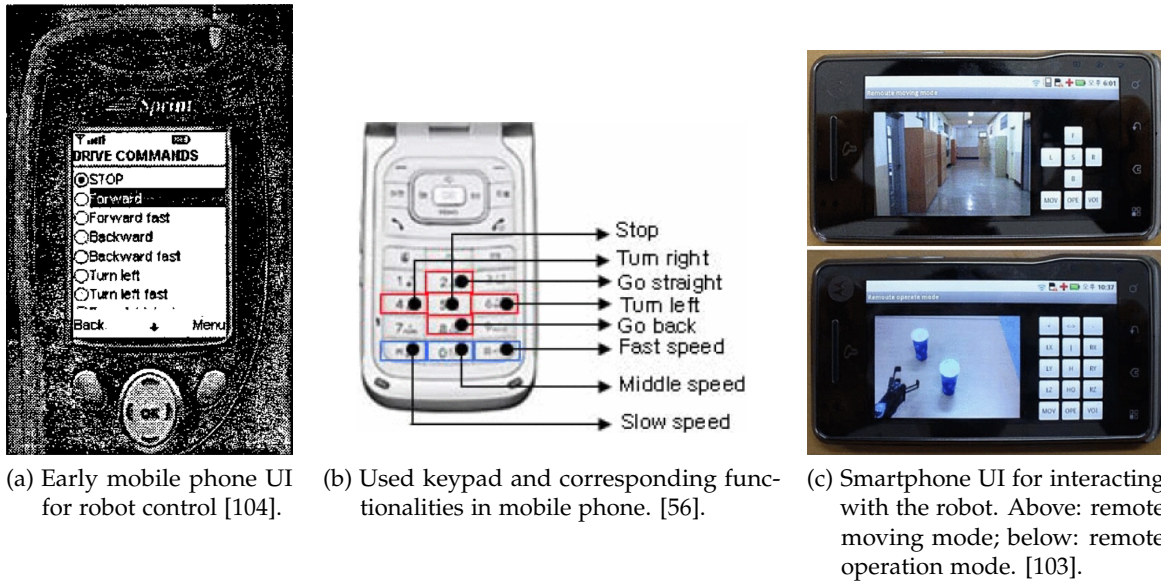


Figure 2.5.: Mobile phone/Smartphone interfaces for controlling robots.

Phone-Robot Data Transmission

There are several methods for data transmission between a mobile phone and a robot. One common approach is to utilize the mobile phone’s wireless internet platform to control the robot. By connecting to an IP network through the mobile communication network, the mobile phone can communicate with a server in the IP network that controls the robot [56]. In 2000, Luo et al. argued that robot control based on the IP network primarily relied on the WWW protocol [107]. However, Cho and Jeon noted that at the time (2008), while controlling the robot over an IP network allowed for the transmission of large amounts of data, practical challenges such as high costs and inconvenience when switching network provider persisted [56].

Liu et al. discussed the complexity of internet transmission, which can lead to long delays and data interruptions [105]. In network-based control systems, data delays and confusion arising from routing and network traffic congestion are inevitable and difficult to accurately predict [108, 109, 110]. It is worth noting that in order to enhance the image transmission speed between the smartphone and the robot, Liu et al. chose the UDP service instead of the TCP service in their experiments [105].

Additionally, Cho and Jeon mentioned that the robot can be controlled using Short Message Service (SMS). The user sends an SMS message to an external server connected to the IP network, the server then analyzes the message content and sends the control signal to the remote robot based on a specific robot control protocol. This transmission method is relatively inexpensive and supported by most mobile phones and network providers. However, the high latency of SMS messages makes continuous control of the robot impractical [56].

There were also attempts to transmit data through Bluetooth [111], WI-FI [105], 485 radio [112], etc.

For the same reasons as Liu et al., in our work, we use UDP as the transmission protocol for communication between the smartphone and the AR HMD. This choice is made because most of our transmitted data consists of time-sensitive but not integrity-sensitive content. Specific details will be discussed in subsection 3.3.2.

2.5.6. Hand Tracking in HRI

Hand tracking has gained significant attention due to its lightweight and intuitive nature in controlling robots. There are various strategies for hand tracking, including the use of data gloves, Leap Motion controller (LMC) ¹¹, and computer vision-based tracking methods. While haptic gloves and depth-based tracking could be promising directions for future work, our focus in this thesis is only on computer vision-based hand tracking methods that do not rely on depth information.

In comparison to using gloves or other hardware, computer vision-based hand tracking is simpler, more natural, and cost-effective [113].

Attempts to control robots by tracking marked hands have been made before. Kofman et al. achieved 6 DoF control of a robotic arm by tracking hands wearing black gloves with markers in specific positions. Schröder et al. developed a real-time hand tracking method using a color glove to drive a robot hand, as shown in ???. They found that using additional depth data obtained from a Kinect camera could enhance pose estimation accuracy. They believed that adding Kinect camera depth information to the database could significantly improve pose estimation accuracy and potentially eliminate the need for the color glove [114].

Gumpp et al. explored a method for controlling a robot hand using markerless hand tracking, as shown in ???. Despite the human hand having over 20 DoFs, to reduce complexity, Gumpp et al. constrained their kinematic hand model to 1 DoF per finger for flexing [115]. Like Stenger et al., they employed a model-based approach and utilized probabilistic algorithms to predict hand poses [stenger, 115].

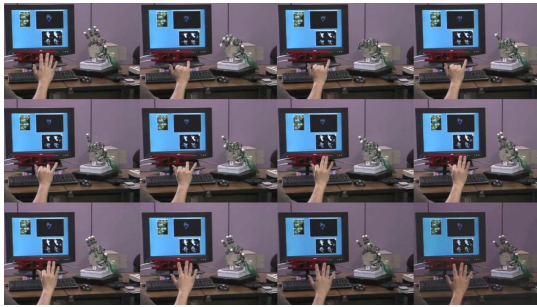
The use of deep learning algorithms for hand tracking is a new trend. Ge et al. proposed a method based on Graph Convolutional Neural Network (Graph CNN) to reconstruct a full 3D mesh of the hand surface, containing richer information about both 3D hand shape and pose. This method can produce accurate and reasonable 3D hand mesh, achieving superior 3D hand pose estimation accuracy [116]. However, such solutions are often not lightweight enough to run on mobile devices [117].

Sreenath et al. explored the use of MediaPipe ¹² to track hands from the monocular camera of a smartphone. They employed simple calibration and perspective projection concepts to obtain the 3D position of the hand relative to the smartphone. Through experiments, they found that this hand tracking solution demonstrated satisfactory accuracy. They further validated the possibility of using hand tracking to control a robot in a simulation, and accurately controlled the end effector movement using the hand tracker [118].

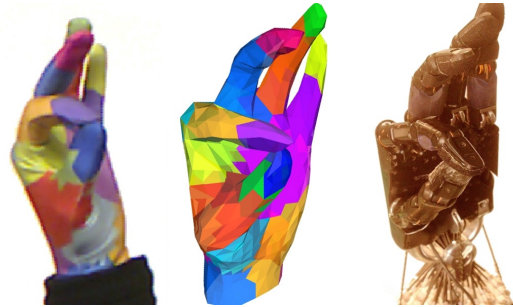
¹¹Ultraleap. <https://www.ultraleap.com>

¹²MediaPipe is an open-source multimedia framework developed by Google. It will be detailed in subsection 3.2.2.

Voigt-Antons et al. compared interaction methods using hand tracking and VR controllers. Through Self-Assessment Manikin (SAM), they found that participants felt a higher valence but lower arousal and dominance with hand tracking. In the grabbing task, hand tracking provided a more realistic experience, resulting in a higher sense of presence. In terms of system usability, assessed using the System Usability Scale (SUS), hand tracking proved to be significantly more usable in the typing task [119].



(a) Markerless real-time hand tracking for robot hand control [115].



(b) Hand tracking using a color glove for robot hand control. [114].

Figure 2.6.: Vision-based hand tracking for robot hand control.

In our development, we utilized real-time hand tracking based on a monocular RGB camera, users do not need additional hardware for controlling. Similar to Gump et al., we constrain the degrees of freedom for each finger to 1 for flexing in order to reduce complexity and increase consistency. While haptic gloves and depth-based tracking may offer better results in the future, they are not the focus of our work.

2.6. Embodiment

Embodiment is a focal point in this paper. Our work sought to realize embodiment during the teleoperation of Roboy. In this section, we delved into the definition and concept of embodiment, its role in robotics and HRI, as well as some related experimental studies.

2.6.1. Definitions and Concepts

The definition and concept of "Embodiment" encompass various aspects [120]. However, its precise definition and impact on Human-Robot Interaction (HRI) remain elusive [121]. Often linked to cognitive functioning, the concept of "Embodiment" carries significant implications for understanding human cognitive development and informing research in the field of robotics [120, 121].

A fundamental concept in cognition is the inseparable interplay between the mind and body, where cognition relies on its physical experience [122, 123, 121]. The notion of "Embodied cognition" posits that cognitive functions are grounded in the physical world, specifically in

the utilization of the body, sensorimotor system, and their interaction with the surrounding environment [120].

The concept of embodiment has expanded to include computing machines and their role in the world [121]. Pfeifer and Scheier [124] provide the following definition:

“Embodiment: A term used to refer to the fact that intelligence cannot merely exist in the form of an abstract algorithm but requires a physical instantiation, a body. In artificial systems, the term refers to the fact that a particular agent is realized as a physical robot or as a simulated agent.”

2.6.2. Embodiment in Robotics and HRI

In addition to various cognitive-focused concepts of embodiment, there are also those that encompass the control of an agent’s sensorimotor abilities [120]. In the realm of humanoid robotics, there is a prominent issue regarding the grounding and localization of sensorimotor and cognitive capabilities that are inherent to humans, including how humans naturally comprehend robots [121]. Haans et al. propose that the telepresence experience arises from the embodiment of the human user [125].

The consensus among roboticists is that incorporating human-like form and functionality in robot design can enhance human-robot interaction, leveraging people’s familiarity with interacting with one another [126, 127]. This includes features such as facial expressions and physical humanoid characteristics. While much research on the embodiment of artificial cognitive systems has focused on the external aspects of robots, Stapleton’s work highlights the growing inclusion of external design and control systems in achieving true cognition [128]. According to Miller and Feil-Seifer, the ultimate objective of embodiment research is to establish a theory that elucidates the qualities of robots that foster positive perceptions among people [129], thus promoting long-term cooperation with robots.

In the field of robotics, researchers frequently aim to replicate human perceptual, cognitive, and sensorimotor functions through technical means, drawing inspiration from these functions for robotic implementation [120]. Experimental studies have further contributed to a clearer understanding of how such implementation should take shape, including the determination of appropriate forms for manipulators, which can range from simple graspers to more complex structures resembling the human hand [130].

Furthermore, there are concepts of embodiment that consider fundamental aspects, including the use of "intelligent" body forms and materials. These traits are believed to have evolved phylogenetically in response to specific environments and ecological pressures, often paralleled by the development of corresponding peripheral and central nervous systems [120].

2.6.3. Experimental Studies

Visual-motor synchronization has been shown to induce a sense of embodiment towards virtual avatars [131]. Numerous experiments have demonstrated the phenomenon of "illusory bodily ownership," where individuals can perceive ownership of objects or bodies other than

their own. For instance, in the "rubber hand" illusion experiment, participants' real arms were concealed while a realistic rubber arm was placed in their view. By synchronously brushing both the hidden hand and the rubber arm, participants reported a compelling illusion that the rubber hand belonged to them [132].

Similar experiments have expanded to explore embodiment of the entire body. Participants' real bodies are concealed through head-mounted displays (HMDs), and tactile stimulation is applied to the hidden real body while corresponding visual stimulation is provided through the shifted video image of the body in the HMD. This multisensory conflict can lead participants to perceive the virtual body as their own and falsely position themselves outside the virtual body [133].

Experiments conducted by Gonzalez-Franco et al. demonstrated that the illusion of body ownership is significantly stronger in the synchronous state compared to the asynchronous state. Participants in the synchronous condition also displayed more frequent responses to threatening events generated in the virtual environment than those in the asynchronous condition [131].

In the study by Hapuarachchi et al., participants controlled a virtual avatar in a virtual room to perform tasks. The avatar was divided into a normal avatar and an abnormal avatar with reversed arms. Results showed that both normal and abnormal avatars with inconsistent movement directions could induce embodiment under synchronous and asynchronous conditions. However, the sense of body ownership and agency in the synchronous condition was significantly stronger than in the asynchronous condition, suggesting that temporal synchrony plays a crucial role in inducing the illusion of body ownership compared to motion direction consistency [134].

Our work aims to synchronize user actions with those of Roboy to the greatest extent possible, while also striving to provide them with a shared vision, in order to achieve the optimal embodiment experience. The synchronization of hand movements is of paramount importance, and we have also attempted various approaches to control the virtual Roboy's hands.

2.7. Immersion and Presence

2.7.1. Definitions and Concepts

Immersion and presence have been long intertwined and explored as interconnected concepts in numerous studies. The concept of presence, according to Lombard and Ditton, has garnered interest from diverse fields such as communications, cognitive science, computer science, engineering, philosophy, psychology, and the arts [135].

The term immersion, as described by Murray in her book *Hamlet on the Holodeck*, has a widely accepted definition:

"Immersion is a metaphorical term derived from the physical experience of being submerged in water.

We seek the same feeling from a psychologically immersive experience that we do from a plunge in the ocean or swimming pool: the sensation of being surrounded by a completely other reality, as different as water is from air, that takes over all of our attention, our whole perceptual apparatus.” [136]

Developed from the description of Murray, the concept of immersion generally refers to the feeling of being of feeling surrounded by something [137, 138, 139]. However, there are various ambiguities associated with the term “immersion” in the context of users’ experience of media, and it is often used interchangeably with “presence” [137, 138, 140, 139].

The concept of presence, defined as the “feeling of being there,” originated from teleoperating systems, specifically the feeling of being in the position of a physically distant robot that the user operates [141, 139, 142]. By the early 1990s, this concept was extended to virtual reality, where users perceive themselves to be immersed not in a remote physical environment, but in a virtual environment presented through a virtual display [141].

Nilsson et al. introduced a taxonomy that includes various concepts of immersion as observed in virtual environments (VE), video games, and fictional literature [137]. This taxonomy divided existing definitions of immersion into three categories:

- (a) immersion as a property of the system,
- (b) immersion as a response to an unfolding narrative,
- (c) immersion as a response to challenges demanding use of one’s intellect or sensorimotor skills.

These three classes can be further categorized into two types: the first is an objectively measurable property of the system and the last two are based on the user’s subjective perception and experience of the events in the virtual world. Most of the existing conceptual categorizations of immersion align with either of these two descriptions, where immersion is seen as either an objective property of the system or a subjective state of the user.

2.7.2. Factors Affecting Immersion

As reflected in Brown and Cairns’ investigation, the users share a common concept of immersion in games, but it is not a static experience but rather relates to the level of engagement in the game. The authors classified game immersion into three levels of engagement: “engagement,” “engrossment,” and “total immersion.” They equated total immersion with the term presence. The authors also acknowledged that usability issues can impact the immersive experience; however, they do not discuss the threshold for such flaws in their work. Given that people invest money in these games, they may invest more effort in learning the game controls and tolerate certain usability problems [140].

The significance of immersion in the game experience is also mentioned by Ermi and Mäyrä. They also highlighted that while the concept of immersion is widely used in discussions about digital games and game experience, its specific application methods remain uncertain and

vague. They note that the term "presence" originally emerged in the context of teleoperation [138], whereas "immersion" is more commonly employed in the context of video games.

Some game reviews suggest that immersion in games can be influenced by factors such as the realism of the game world and atmosphere sound [138]. However, McMahan argued that many scholars and scientists share a perspective that total photo- and audio- realism are not necessary for viewers to have a sense of immersion [139]. Similarly, Ermi et al. agreed that while the audiovisual implementation of a game may contribute to the immersive experience, they are not the sole or most significant factors [138]. McMahan also outlined three conditions necessary for creating a sense of immersion in digital games: (a) the conventions of the game matching the user expectations, (b) meaningful things to do for the player, and (c) a consistent game world [139].

Quantifying immersion poses a significant challenge. Achieving a system that is objectively more immersive requires careful consideration of several factors, including:

- (a) sensory aspects such as visual and auditory fidelity,
- (b) behavioral fidelity of simulated objects,
- (c) minimizing display lag and system latency,
- (d) maximizing tracking coverage,
- (e) accounting for environmental factors such as temperature, air flow, gravity, and sensory isolation from the real world,

and many more [143].

2.7.3. Separation of Immersion and Presence

Slater argued that "immersion" and "presence" should be regarded as separated terms, since they are logically separable [144]. To alleviate any ambiguity, Slater maintains the term "place illusion" to denote presence, while also introducing the term "plausible illusion" to describe the illusion of feeling that what is virtually happening is really happening. Slater et al. defined the concept of immersion as a description of characteristics of a system [144, 143]. Later Slater summarized that the quality of experience is influenced by various parameters, including graphics frame rate, overall extent of tracking, tracking latency, image quality, field of view, visual quality of the rendered scene, dynamics, and range of sensory modalities accommodated [143]. A system is deemed more "immersive" than another if it surpasses it in at least one of these characteristics [143]. According to his viewpoint, "immersion" should exclusively pertain to what the technology delivers from an objective point of view, specifically referring to the extent to which a system provides displays and tracking that preserves fidelity in relation to the equivalent real-world sensory modalities [141]. This definition aligns with the taxonomy presented by Nilsson et al., where immersion is considered a property of the system [137]. Slater further distinguishes presence as the subjective human response to immersion. While Slater argued that "immersion" can, in principle, be objectively assessed,

individuals may experience different levels of presence even when exposed to the same immersive system [144, 143]. Correspondingly, the same level of immersion may be yielded by different immersive systems for different individuals.

Slater argued that immersive systems can be described in terms of the sensorimotor contingencies (SC) they support. SC refers to actions performed in order to perceive, such as moving the head or bending down to see underneath something [145]. If the device does not support head tracking, then head movement will have no effect, because such movements have no effect on perception. Whereas a system that supports SC close to physical reality may give the user the illusion that they are located inside a rendered virtual environment. This illusion has been referred to as telepresence or presence [144, 146, 147]. In addition to matching the display and interaction capabilities to the requirements of the human perceptual and motor systems, increasing realism is also a way to induce presence [143].

3. Approach

3.1. Introduction

3.1.1. Overview

The core focus of our project implementation lies in software development, which includes two main components: a smartphone application and a simulation application of virtual Robody and clinical environment simulation based on AR HMD ¹. The former is a project developed from scratch, serving as a comprehensive interface for controlling Robody's actions. On the other hand, the latter builds upon previous works, simulating the real-world perceived by the Robody's physical entity. These two components work cooperate, offering users multiple ways to control Robody. Through these interaction methods, a comprehensive control experience that integrates various control modes and seeks lightweight realization of embodiment is created, in order to fulfil different needs in diverse scenarios. This project opens up broader possibilities for the practical application of controlling the Robody's physical entity.

3.1.2. Previous Works

Before our work, there were already some impressive pioneering works related to controlling virtual Robody in AR environment. These studies were conducted by Kawabata and Gao and primarily focused on real-time mapping of virtual hands to user hands, interaction methods, and multi-perspective observation techniques [148, 149]. These prior works have provided valuable references and an important foundation for our project.

Kawabata's Work

Kawabata's work marks the starting point of this project. She built a system from scratch that utilizes Unity's Inverse Kinematics (IK) technology. This system enabled the real-time mapping of user's hands' position in 3D space to the virtual Robody's hands position in the virtual environment based on the movement of the Magic Leap 1 controller. Users could use the controller's trigger button to perform simple gripping simulations with Robody's hand for interaction with objects in the scene. Additionally, she explored the application of hand tracking in controlling Robody's hand [148]. Kawabata's work laid the foundation for this project, establishing a framework for interaction between the virtual and real worlds.

¹Referred to as "AR environment" in the subsequent text.

Gao's Work

Gao's work represents a further development and improvement of Kawabata's work. Building upon Kawabata's foundation, Gao introduced a more intuitive user interface, allowing users to switch between different Robody control modes more easily. He added multiple viewing options, enabling users to observe and operate Robody from different perspectives, enhancing spatial awareness and self-awareness of their own pose. Gao introduced a more complex scene and more interactive objects, allowing users to control the Robody in various virtual scenarios [149]. His work further enhanced the practicality of the control system, providing users with more choices and a richer experience, making it more suitable for a wider range of applications.

Devanthro's VR control solution

Devanthro has already developed a VR solution for achieving remote embodiment of Robody. This is a well-established and mature solution that has been successfully implemented on physical Robody.

In this solution, the user wears a VR HMD, which displays Robody's perspective. The user controls Robody by rotating the head to make Robody's head follow the head movement. Each hand of the user holds a VR controller, responsible for controlling one of Robody's hands. When the user presses the trigger button on the controller with the index finger, Robody's hand tracks the user's hand movements synchronously. Users can also control Robody's movement using the joystick on the controller.

3.1.3. Goals

Before looking at the implementation details, it is necessary to summarize the implementation goals. Since our project consists of two main components, our implementation objectives can be roughly divided into two parts: one for the smartphone application and the other for the AR environment. Each part has its unique focus and requirements, but they also share some common objectives.

Smartphone Application

In this part, our primary goal is to design and develop a user-friendly smartphone application that provides a straightforward, intuitive, and convenient way to control and monitor Robody. Additionally, it should delicately integrate with the AR environment to achieve more effective interaction with the virtual Robody. The following outlines our detailed goals and specifications:

1. **User-Friendly Interface:** We aim to create an intuitive and user-friendly interface to ensure that users can easily understand and operate the application, allow them to effortlessly interact with Robody and access information. This includes a clear menu structure, easily understandable icons and control buttons.

2. **Data Exchange with AR Application:** The application need to be capable of real-time data exchange with the AR simulation program through some communication channel. This ensures a seamless connection and interaction between the smartphone application and the AR environment. This includes the transmission of pose data, video stream, as well as task commands.
3. **Real-Time Feedback on Robody's Status and Perspective:** Users should be able to receive real-time updates on Robody's motion status, sensory information, and perspective. This enhances the user's sense of control and awareness, enabling more effective control of Robody.
4. **Task Assignment:** The application should allow users to assign tasks to Robody, enabling Robody to autonomously complete specific tasks or activities. This enhances Robody's autonomy and practicality, reducing tedious and repeated work in certain scenarios.
5. **Hand Control** The application should also enable users to control Robody's hand movements using smartphone. This provides users with more control options when not relying on AR device controllers, allowing them to intuitively control Robody's hand movements.

AR Environment

In this part, our goal is to build upon previous works and enhance an AR experience that is integrated with the smartphone application, allowing users to control Robody's movements in the AR environment without relying on AR device controllers (and interact with the virtual clinical environment). The following outlines our detailed goals and specifications:

1. **Enhanced Virtual Environment:** The virtual environment displayed in the AR headsets should include more objects to better simulate complex clinical settings in the real world, enhancing user immersion and experience.
2. **Simulation of Virtual Robody:** The virtual Robody in the AR environment should have a reasonably natural movements and behaviors, enabling users to achieve a better sense of embodiment (SoE).
3. **Layered User Interface:** Users should be able to easily switch between different control modes and perspectives for the most comfortable and adaptable user experience.
4. **Integration with Smartphone Application:** The AR environment should be capable of real-time data exchange with the smartphone application through some communication channel, allowing for the reception of commands from the smartphone and the transmission of pose information and video streams.

3.1.4. Hand Control Strategies for Robody

In previous works, Robody's hand movements were primarily controlled using the controller that came with the AR headset. The controller achieved three-dimensional motion tracking with 6 DoFs through high-precision motion sensors and the assistance of the AR HMD device². However, in practical scenarios, carrying a separate controller is not convenient or flexible. The controller has a single function, serving only to control Robody in AR environment, and becomes a burden for users when not interacting with AR. Moreover, its relatively large size and irregular shape make it inconvenient for everyday carry. Therefore, our aim is to effectively control Robody's movements without relying on a controller.

But due to various limitations in hardware and software, we currently do not have a perfect solution that can simultaneously achieve embodiment, portability, and precise control. All the current control strategies for Robody's hands have their own advantages and limitations. Only a relatively balanced solution can be found for different scenarios.

Haptic Glove-Based

The most precise and embodiment-oriented solution is to use haptic gloves currently under development to control Robody's hands. Haptic gloves can provide precise hand tracking and haptic feedback, allowing for maximum embodiment and precise control. However, the drawback of haptic gloves is that they are, at this stage, very bulky and not easily portable. Moreover, when wearing the gloves, users cannot freely perform other tasks as their hands are occupied, significantly limiting the practical use of the haptic gloves in various clinical scenarios.

Hand Tracking-Based

Another solution is to control Robody using computer vision-based hand tracking. Hand tracking allows users' hands to be completely free from hardware constraints, requiring no controllers other than a head-mounted AR display. Users can intuitively control the robot anytime and anywhere. This is the ideal lightweight solution, but with current technologies, there are still many limitations. This control strategy heavily relies on the quality of the camera feed. Dim or uneven lighting, low video resolution, and low frame rates can lead to instability in hand tracking. Existing computer vision algorithms for hand tracking are based on finger recognition, so hand occlusion, including self-occlusion between fingers, can severely affect accuracy and stability. Additionally, while this hands-free control method offers maximum freedom, it lacks haptic feedback, negatively impacting the sense of embodiment.

Smartphone-Based

Using a smartphone is a compromise between the two aforementioned strategies. The motion sensors commonly found in smartphones, especially the gyroscope, can serve as a reasonably

²ML1 Control Overview. <https://www.magicleap.care/hc/en-us/articles/360033747511--ML1-Control-Overview>

reliable tool for measuring the phone’s orientation. However, due to the limitations of the motion sensors in smartphones, particularly the accelerometer, they are not capable of accurately tracking the phone’s translation in three-dimensional space. With only 3 DoFs provided by gyroscope, users need to interact with the touchscreen to gain additional DoFs to control the hand movement in 3D space. Smartphones can thus offer fairly reliable and precise hand pose control while sacrificing some freedom of hand movement. However, this control method cannot intuitively map the user’s hand movements to Robody’s hand movements, greatly diminishing embodiment and requiring a learning curve to become proficient.

While relying solely on motion sensors in smartphones may not yield precise 6-DoF device tracking, with the assistance of the rear-facing camera commonly equipped in smartphones, we can take advantage of computer vision algorithms, such as simultaneous localization and mapping (SLAM), for device motion tracking and pose calibration. With the assistance of current AR solution frameworks, smartphones can provide 6-degree-of-freedom motion tracking through sensor fusion. This allows for intuitive mapping of the user’s hand position to Robody’s hand position, compensating for the lack of translation tracking in the previous strategy and enhancing embodiment.

Additionally, the vibration function in smartphones can offer users haptic feedback, further enhancing embodiment. However, controlling Robody’s finger movements through a smartphone, similar to using a controller [Gao], cannot be easily realized.

However, in addition to the aforementioned features, smartphones possess a unique and significant advantage: their ubiquity. Smartphones, as devices with considerable computing power and output capabilities, are well-qualified to perform many computational tasks. They are used as a daily necessity by a vast majority of people, eliminating the need for additional hardware. They are equipped with reliable sensors that can perform device pose tracking at a relatively low cost, without external help. Smartphones commonly feature multiple communication channels, allowing for effective communication with other hardware.

Strategy	Estimated Level of SoE	Advantages	Disadvantages
Haptic gloves	+++	Best SoE	Cumbersome, expensive hardware
Hand-Tracking	++	Light-weight	Very sensitive to interference
Smartphone without camera input	+	Easy access, robust against interference	Limited SoE
Smartphone with camera input	++	Easy access, better SoE than that without camera input	Relatively sensitive to interference

Table 3.1.: Comparison of different hand control strategies.

3.2. Tools and Technology

3.2.1. Hardware

The main hardware components used in our implementation include smartphones, an AR HMD, and a wired webcam. Given the real-world context, we made the following hardware device selections during the development and testing phases:

Smartphone: Apple iPhone 11

We used the Apple iPhone 11 as our smartphone device. The iPhone 11 is equipped with advanced processors, high-resolution displays, sensitive sensors, and powerful graphics capabilities, making it an ideal choice for our application. These features ensure smoothness and responsiveness of the application, and a variety of sophisticated sensors also provide more control possibilities, enhancing the user's interactive experience while controlling Robody.

AR Headset: Magic Leap 1

Considering real-world factors and technical requirements, we have chosen the Magic Leap 1 as our AR HMD device. The Magic Leap 1, released in 2018, is an AR HMD that can perceive the real world environment, track user position and orientation, and use this information to blend virtual content with the real world, providing users with an AR experience. It comes with a controller capable of tracking 6-DoF motion, offering users a robust and intuitive method of interacting with the AR content. While the Magic Leap 1 may no longer be the latest device showcasing cutting-edge technology, it still provides the essential functionalities needed for our project development.

Webcam

When developing features related to hand tracking, we need to obtain the camera data before we can recognize hand poses from the content captured by the camera using computer vision algorithms. Because Magic Leap 1 itself is equipped with cameras for head tracking and its native hand tracking, our most direct option is to utilize the cameras natively embedded in Magic Leap 1. However, due to the outdated API of the Magic Leap 1, accessing its raw camera data directly presents challenges and requires extra effort. Additionally, considering that Magic Leap 1 is not a future-proof solution, we have decided not to specifically implement a method for obtaining its raw camera data. Instead, we have chosen to use an easily accessible webcam for capturing video. Webcam-captured images facilitate our development and testing efforts. Moreover, the associated implementation is not coupled with a specific platform, which enhances future adaptability and improvements.

3.2.2. Software

During our development process, software plays a crucial role. We mainly rely on the Unity engine for most of our development work, including the creation of the smartphone application and the AR environment. Additionally, we utilize a range of supporting software, frameworks, and software development kits (SDKs). These tools collectively form the foundation of our project.

Unity Engine

The Unity engine ³ is the absolute centerpiece of our development. Unity is a widely-used cross-platform game engine for game development and mixed reality applications. It offers a rich set of development tools and resources that help developers to create high-quality games and mixed reality applications. Unity engine has robust graphics rendering and physics capabilities, along with a vibrant development community that provides numerous plugins and resources to accelerate our development process. Unity's cross-platform capability facilitates our development by allowing us to efficiently integrate the smartphone application and the AR environment with each other.

ARKit

ARKit ⁴ is a software framework developed by Apple for creating AR applications on iOS devices such as the iPhone and iPad. It has become a powerful tool for iOS developers to build AR applications.

ARKit, along with its Android equivalent ARCore ⁵ and Vuforia ⁶, provides highly accurate device tracking capabilities based on simultaneous localization and mapping (SLAM). Taking ARKit as an example, it utilizes a technology called visual-inertial odometry, which combines motion sensing hardware data from iOS devices with computer vision analysis of the visible scene from the device camera. ARKit can recognize features in the scene and track changes in the positions of these features to create a high-precision model of device's motion and position ⁷.

With the device tracking capabilities offered by ARKit, it becomes straightforward to create a smartphone application that can perform 6-DoF motion tracking. This capability offers us an important choice in how we control Robody's movement using smartphones.

³Unity. <https://unity.com>

⁴ARKit. <https://developer.apple.com/augmented-reality/arkit/>

⁵ARCore. <https://developers.google.com/ar>

⁶Vuforia. <https://developer.vuforia.com>

⁷Understanding World Tracking. https://developer.apple.com/documentation/arkit/arkit_in_ios/configuration_objects/understanding_world_tracking

MediaPipe

MediaPipe⁸ is an open-source multimedia framework developed by Google, offering a set of tools and libraries for multimedia data processing and analysis. It focuses on computer vision and machine learning tasks such as pose estimation, hand tracking, facial detection, etc. MediaPipe uses a modular design, allowing developers to select and combine different processing modules to build custom media processing pipelines. For example, in our project, we only needed to invoke the hand tracking module to output the three-dimensional coordinates of hand landmarks without requiring the video stream output module, making the development process more flexible.

MediaPipe is not designed to provide high-precision tracking and recognition but is optimized for real-time multimedia data stream processing on mobile devices, making it well-suited for devices with limited computational resources and battery, such as AR HMDs. There is already a mature open-source plugin available for integrating MediaPipe into the Unity Engine, significantly simplifying our development and debugging processes.

We primarily utilized the hand landmarks detection module of MediaPipe, which utilizes an Machine Learning (ML) pipeline consisting of two packaged models: a palm detection model and a hand landmarks detection model. The palm detection model locates hands within the input image, while the hand landmarks detection model identifies specific 2.5D hand landmarks within the cropped hand image defined by the palm detection model [117].

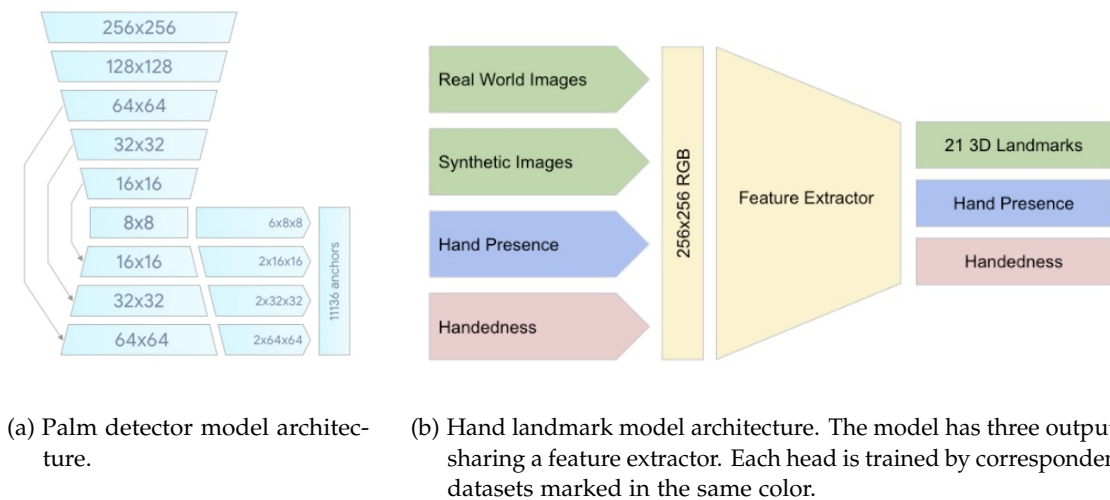


Figure 3.1.: MediaPipe hand tracking solution architecture [117].

By employing specific techniques, palm detection achieved an average accuracy of 95.7% [117]. The hand landmarks detection model model was trained on approximately 30K real-world images and several rendered synthetic hand models imposed over various backgrounds.

⁸MediaPipe. <https://developers.google.com/mediapipe>

3.3. Implementation

3.3.1. Project Setup

Our project involves various hardware and different platforms. While we have the assistance of Unity Engine, setting up the two projects remains a complex task. This complexity arises from compatibility issues between different software versions and varying development tools. Additionally, refining and making adjustments to the works done by Kawabata and Gao also demands a significant amount of time and effort.

System Overview

Due to compatibility issues, we did not choose the latest version of Unity Editor in our project development. The AR environment development is based on Unity Editor 2020.3, which is also the last official version supporting Magic Leap 1. However, to enable smartphone motion tracking, we needed the ARKit versions providing device tracking feature. The earliest Unity Editor version supporting these ARKit versions was 2021.3. Therefore, we selected Unity Editor 2021.3 as our choice for developing the smartphone application.

Project Transition and Refinement

The early development of the AR virtual environment was built upon the work of Kawabata and Gao. By modifying their Unity project and adding new features, we were able to quickly create a functioning prototype that facilitated communication between smartphones and AR devices.

Although this allowed us to rapidly enter software development, there were significant limitations in building upon their project's foundation. Firstly, due to differences in project focus, many features in their work were not needed for our project, such as the miniature world perspective. Secondly, since they had independently completed their projects, each person used different third-party assets, resulting in redundant assets within the project, some of which had complex dependencies on core content. Lastly, developing for the Magic Leap 1 in Unity required complex configuration with the installation of numerous SDKs and packages. After both of them made modifications, there were some dependency conflicts in the Unity project, which would also hinder future development.

Considering all of the above, we decided to create a new Unity project and meticulously document and organize all relevant SDKs, packages, including the Unity Editor version, while retaining only the assets in use. This would facilitate better workflow for future work. We continued to use MRTK for Unity but abandoned the Magic Leap Toolkit, as the latter had not been updated for the past three years and had irreconcilable conflicts with various versions of MRTK for Unity. As our control mode in development had made controllers optional, the previous Robody hand control logic, which was centered around tracking controllers in their work, was no longer applicable. Therefore, we rewrote a new hand control logic to replace the previous implementation.

3.3.2. Device Communication

In our project, it is essential for smartphones and AR HMDs to collaborate, thus the data transmission between them is crucial. Smartphones and AR HMDs typically offer multiple data transmission channels, such as Bluetooth, Wi-Fi, and NFC. Among these options, Wi-Fi connectivity is supported by most devices. It is not only convenient for development and debugging but also offering an extended transmission range, making it an ideal choice. Our device communication system is illustrated in Figure 3.2.

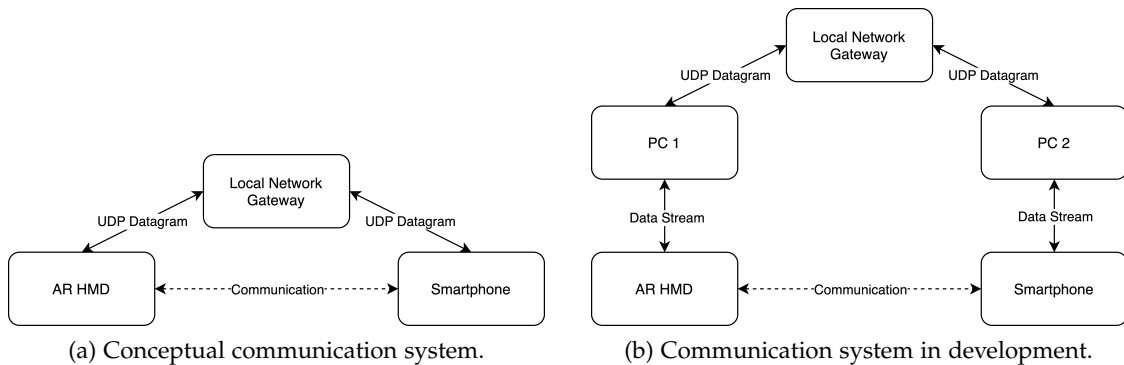


Figure 3.2.: Device communication system.

Transmission

In our design, smartphones and AR HMDs collaborate at the same location without the need of being on separate local networks. To facilitate this collaboration, our implementation defaults to placing both devices within the same local area network. The advantage of this approach is that it allows us to avoid complex network address translation (NAT) traversal mechanisms, avoiding time-consuming IP address acquisition for the other device.

To optimize data transmission efficiency, we have chosen the connectionless UDP as the transmission protocol. UDP is renowned for its fast transmission speed. It is highly suitable for applications that require real-time responsiveness and data communication. UDP ensures that the delay in data transmission between smartphones and AR HMDs is minimized as possible, providing users with an immersive experience when controlling Robody in real time.

UDP is also known for its unreliability. In our project, most of the data transmission does not need every data packet to be delivered intact, such as video streams and location information. The loss of some data packets does not significantly impact the overall performance or result in failures. However, for certain data, such as confirmation messages that require reliable transmission, we ensure the integrity of this information by returning acknowledgment messages to confirm their successful delivery.

Packet Structure

In the application layer, the basic network message is conceptually divided into two parts: the header and the payload. The header includes information about the source and destination IP addresses for identification use, a flag to determine whether reliable transmission and acknowledgment are required, message type, and timestamp. The payload contains a string of the content to be transmitted. Binary data is serialized into a string format when transmitted.

Due to the limited size of a single UDP datagram (8-byte header + 65,527 bytes of data), data exceeding this capacity, such as video stream data, requires additional data structures to segment it into smaller parts for transmission across multiple UDP datagrams.

3.3.3. User Interface

The design of the user interface (UI) is crucial for the user's experience and efficiency. We need to ensure that the UI is simple, intuitive, and smooth, minimizing the user's learning curve and operational difficulties.

Smartphone Application UI

In our smartphone application, we have adopted a common layout scheme, similar to many smartphone applications, which involves placing a tab bar at the bottom of the screen. This tab bar allows for quick switching between different interfaces, each serving a distinct purpose. Our tab bar provides access to three main pages: monitoring, control, and tasks.

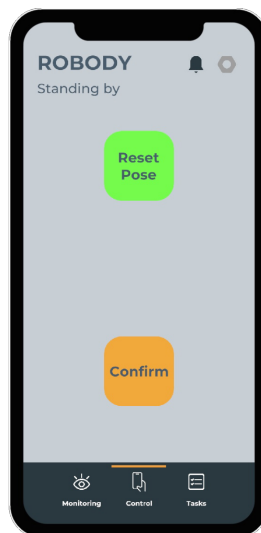


Figure 3.3.: Control page interface. This page offers a confirm button and a pose reset button.

In the monitoring page, users can view Robody's real-time perspective and its position on a mini-map. This feature provides users with a clear understanding of Robody's current status,

allowing them to stay informed about Robody's location and the surrounding environment at any time.

The control page consists of two buttons: a confirm button and a pose reset button. These buttons are used to send confirmation commands and reset the device's orientation and position. The confirm button serves as an alternative to the controller's Trigger button in absence of AR controller (see Figure 3.4). The pose reset button is used to reset the device's initial orientation and position, ensuring that users can keep using the device tracking feature properly after changing their position and orientation. We will discuss the functionalities of this interface in more detail in the following parts.

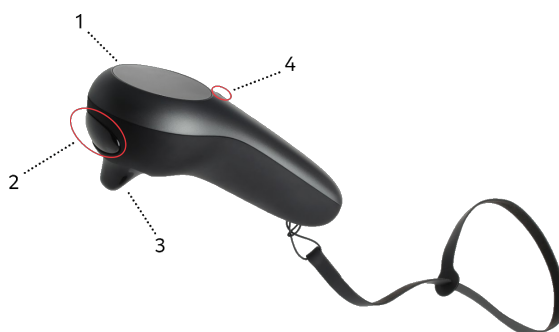


Figure 3.4.: Magic Leap 1 controller (Magic Leap 1 Control). 1: Touchpad; 2: Bumper; 3: Trigger; 4: Home Button. The trigger is the default button for selecting objects.

Within the tasks page, users can access a range of pre-defined tasks. Clicking on any task opens a detailed information popup, where users can set parameters and details for the task. Subsequently, the user can assign the task to Robody for execution. The details about task assignment will be discussed in subsection 3.3.6.

AR Environment UI

In previous works by Kawabata and Gao, users controlled Robody and interacted with the UI through AR controllers in the AR environment. One of the key focuses of our work is to break free from the constraints of AR controllers and rely only on smartphones and AR HMDs for controlling Robody. Therefore, we also needed a UI interaction method that does not depend on controllers and can be used in their absence.

In subsection 2.4.2, we discussed the gaze and touch technology. This mixed interaction approach across devices provides us with valuable insights and inspiration. Our solution is to replace the trigger button, which AR controllers typically use for confirming object selection, with a touchscreen button on the smartphone, as shown in Figure 3.3. We also change the method of aiming and selecting objects from using a controller to aiming at the object in the center of view. When a user wants to select a specific object, they simply need to turn their head to aim at the object with the help of the cursor in the center of view and then tap

the confirmation button on their handheld smartphone screen to select it. The selection is illustrated in Figure 3.5.

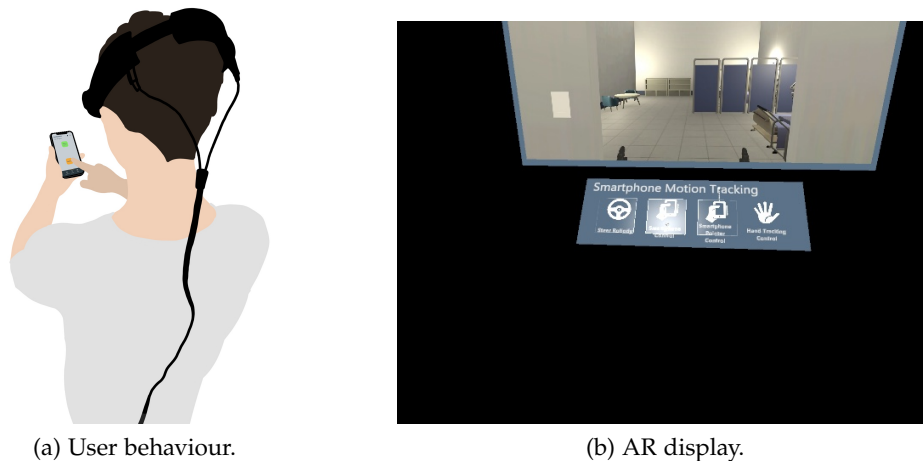


Figure 3.5.: Illustration of main menu button selection.

In previous works, when users entered the AR environment, they would immediately enter Robody's first-person perspective. We divided the AR environment into two scenes: users would not directly enter the virtual clinical environment simulation but would first arrive at a main menu scene. Here, users could choose the control mode they wanted to use by pressing the corresponding UI button using the UI interaction method described earlier. The available control modes are displayed in a row in the main menu scene, allowing users to clearly see different choices. Above the buttons, there is a screen where users can see Robody's real-time perspective. Another advantage of the main menu scene is that it is convenient for adding extended content, such as displaying patient information. The main menu UI is shown in 3.6a.

After choosing the control mode, users would be "teleported" to the second scene: where Robody is located. Here, users share Robody's first-person perspective. If they choose to control Robody's hands through the smartphone, the smartphone touchscreen would overlay the hand control interface. This interface provides the buttons and sliders needed to control Robody's hands. When entering the control mode, Robody's hand control would not take effect immediately but remain in a frozen state. Users need to manually press the "Unfreeze" button on the smartphone screen for Robody's hand control to become active. This is to prevent unexpected movements of Robody's hands before the user is well-prepared. Users can freeze Robody's hand control at any time and return to the main menu. Users can also choose to switch the controlled hand when Robody's hands are frozen.

In this scene, there is also a menu floating above the user's field of view, consisting of four buttons. These buttons can be triggered through gaze with dwell-time (set to 1.5 seconds by default). These buttons include: 1. a main menu button allowing users to return to the main menu after being triggered; 2. a pair of buttons to freeze and unfreeze Robody's

hand movements, enabling users to freeze or unfreeze the hand's movement when using the smartphone is impractical, such as during hand tracking; 3. a button to switch between controlling the left and right hands, which can only be triggered when Robody's hand are frozen to prevent unwanted hand actions.

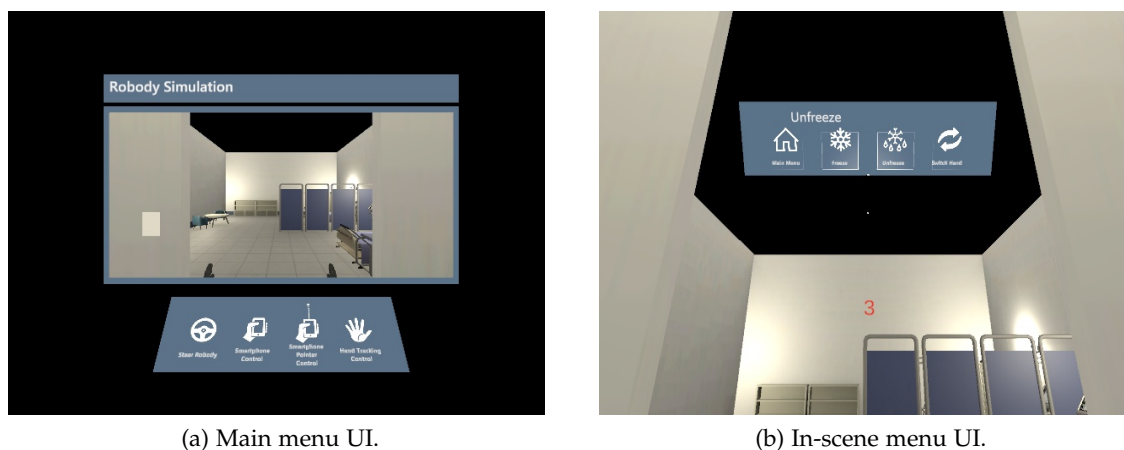


Figure 3.6.: Menu UIs.

3.3.4. Workflow

When utilizing our control system, users need to follow a specific workflow. Firstly, the user needs to connect the smartphone and the AR HMD to the same local area network (LAN) as this is the foundation of the communication between the two devices. Subsequently, the user needs to simultaneously open the smartphone application and the AR application. The user will initially see the main menu in the AR HMD, while the smartphone displays the control interface.

While in the main menu, user actions are unrestricted, allowing them to freely switch between monitoring or task assignment pages in the smartphone application. If the user wishes to employ a smartphone-based control mode, they need to calibrate the smartphone's orientation and position before entering the control mode.

To calibrate, the user needs to hold the smartphone upright in front of their chest with the screen facing them while in the main menu. When the "pose reset" button on the control page is highlighted, they should press it to reset the smartphone's orientation and position. At this point, the smartphone will record the current position and orientation as the correct upright position in front of the chest, and all subsequent movements and rotations will be based on this pose. The calibration operation is illustrated in Figure 3.7.

In the main menu, users can also select the steering control mode, in which they can control Robody's movements using directional buttons in the smartphone application. Upon selecting this mode, similar to hand-tracking mode, users directly enter the virtual scene for control and hand control will be disabled. Users can also steer Robody in smartphone-based control

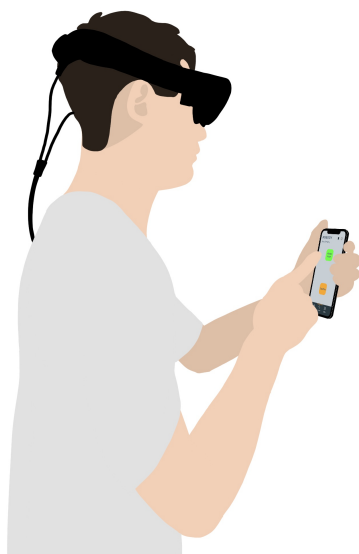


Figure 3.7.: Illustration of calibration operation.

modes. When users have frozen hand control, the smartphone screen provides directional buttons to allow users to change Robody's position.

In the smartphone-based control mode, as mentioned in subsection 3.3.3, users must first unfreeze Robody's hand control before being able to control Robody's hands. To return to the main menu interface, users must first freeze Robody's hand control and then exit the control mode. The same applies to hand tracking mode, where users' hand movements are frozen when entering the control mode. Users need to unfreeze hand movements through the button above their field of view to activate hand tracking.

After returning to the main menu, users have the option to select any control mode once more. The whole workflow is visualized in Figure 3.8.

3.3.5. Monitoring

In Gao's work, users have the option to select a monitoring mode in the AR environment. In this mode, the user's perspective in the AR HMD is "teleported" to a predefined location and held there for a specified period for the user to monitor the (virtual) clinical environment. However, in real-world applications, this monitoring approach relies on external devices, such as cameras installed at specified locations. Furthermore, this approach lacks flexibility as users cannot easily change the monitoring location.

With the assistance of a smartphone, we introduce a different monitoring strategy: users can directly access Robody's vision in the AR environment from the main menu or the smartphone application. Robody's perspective is transmitted in real-time to the smartphone over the local network and displayed on the monitoring page, as shown in Figure 3.9. Additionally,

3. Approach

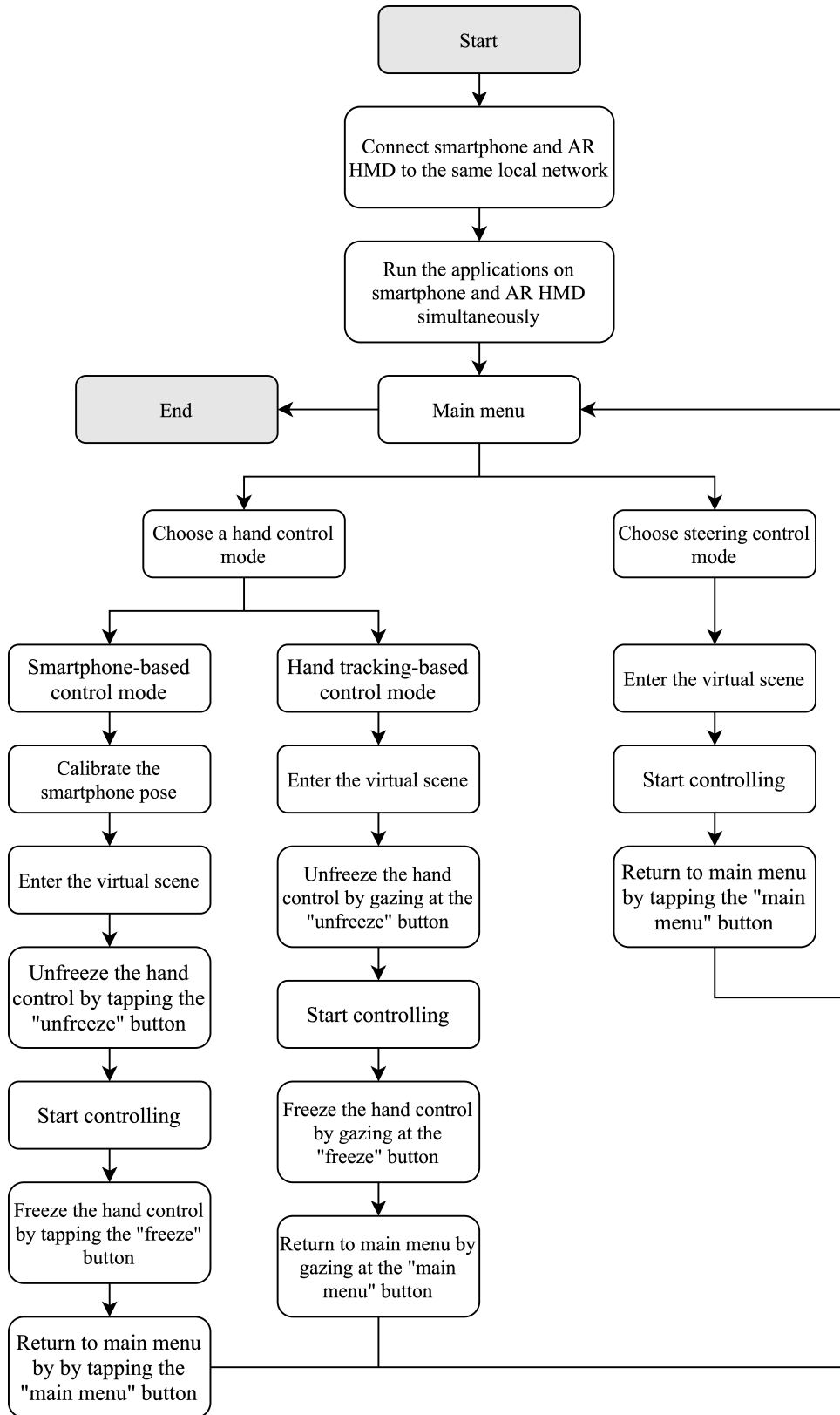


Figure 3.8.: Workflow of the control system.

a pre-made bird's-eye-view minimap of the simulated environment is displayed below the monitoring graphic, and Robody's real-time position is transmitted to the smartphone and shown on the map. This allows users to gain a comprehensive understanding of Robody's location and surroundings.

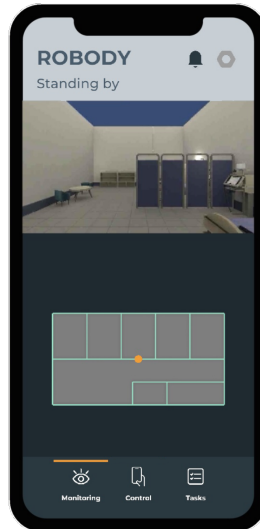


Figure 3.9.: Monitoring page interface. In the upper part of the screen, the simulated scene is displayed, which ideally streams the real-world perspective of Robody in practical applications.

3.3.6. Task Assignment

Gao's work emphasized the significance of autonomy of the Robody in various contexts. In his future work prospects, he highlighted the importance of allowing the robot to handle low-risk tasks independently, thus alleviating the operator's workload. Gao's user research further revealed that the users expressed a desire for autonomous movement.

Therefore, we designed a task assignment system. This system allows users to operate on their smartphones, where they can choose tasks from a predefined set of tasks that they want Robody to perform autonomously. After configuring specific parameters, they can then assign the task to the virtual Robody within the AR environment.

These tasks include displacement and patrol. In the displacement task, the operator can specify the exact location for Robody to go by clicking or dragging on the mini-map. After the task is assigned to the virtual Robody, it will head towards the specified location. In the patrol task, the operator can define a path by selecting multiple waypoints on the mini-map and then assign the task to the virtual Robody. It will then patrol along the designated path repeatedly.

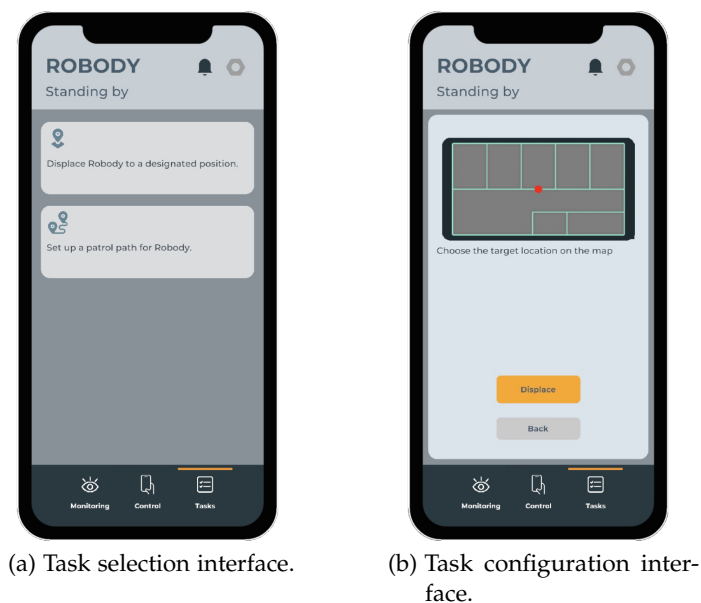


Figure 3.10.: Task assignment page interface.

3.3.7. Hand Control

In the previous sections, we have examined various feasible Robody hand control strategies. Among them, visual-based hand tracking and smartphone-based control strategies are our primary focuses. We implemented three distinct methods for controlling Robody's hands, which will be discussed in detail in the following parts.

The basic idea is that we use different interaction methods to move the IK target within the AR environment, and the Robody in the AR environment calculates the rotation angles required for each joint to make its hand as close as possible to the IK target with the help of the IK solver. This allows us to control the Robody's hand.

Hand Tracking-Based on Monocular RGB Webcam

Through monocular RGB webcam, coupled with the hand tracking solution provided by MediaPipe, as introduced in subsection 3.2.2, we realized high-precision tracking of the user's hand's pose and mapped the hands' pose onto the virtual Robody's hands in the AR environment. This means that Robody can move its arms according to the user's hand movements captured by the webcam. The concept is depicted in Figure 3.11.

In this process, we pass the video stream data obtained from the webcam to the MediaPipe plugin embedded in Unity, which processes each frame to obtain the 2.5D coordinates of 21 landmarks (42 landmarks for 2 hands) on the user's hands.

MediaPipe cannot measure the depth of the hand (the distance between the hand and the camera). The distance between the hand and the camera is reflected in the size changes (the detected hand model is larger when the hand is closer to the camera and smaller when

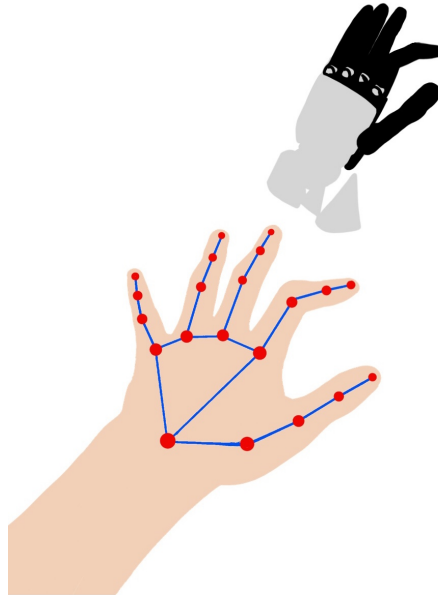


Figure 3.11.: Hand tracking control mode concept.

farther away). To calculate the depth of the hand and obtain the complete 3D position of the hand, we predefine a standard hand size. By computing the ratio between the distance of the tracked landmarks 0 and 5 (see in Figure 3.12) and the standard hand size, multiplied by a predefined factor, we can obtain information about the depth changes of the hand. The concrete value of the hand's depth is controlled by the predefined factor and can be adjusted to provide the better tracking performance for specific users.

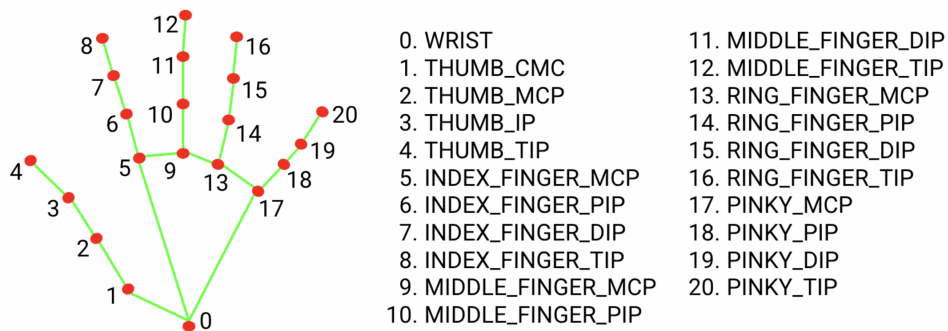


Figure 3.12.: MediaPipe hand landmark model⁹. The orientation of the palm is calculated from the landmarks 0 (WRIST), 5 (INDEX_FINGER_MCP), 17 (PINKY_MCP).

We derive the hand orientation information from three landmarks on the palm, and use it in conjunction with the position of the wrist to drive the IK targets for the Robody's hands:

⁹Hand landmarks detection guide. https://developers.google.com/mediapipe/solutions/vision/hand_landmarker.

$$\begin{aligned}\vec{v}_1 &= l_5 - l_0 \\ \vec{v}_2 &= l_{17} - l_0 \\ \vec{f} &= \frac{\vec{v}_1 + \vec{v}_2}{|\vec{v}_1 + \vec{v}_2|} \\ \vec{u} &= \frac{\vec{v}_1 \times \vec{v}_2}{|\vec{v}_1 \times \vec{v}_2|} \\ \vec{r} &= \frac{\vec{f} \times \vec{u}}{|\vec{f} \times \vec{u}|}\end{aligned}$$

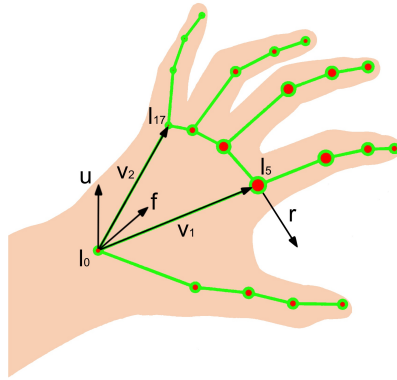


Figure 3.13.: Demonstration of palm orientation calculation.

where l_0, l_5, l_{17} are positions of number 0, 5, 17 landmarks of the detected landmarks, \vec{f} is the normalized forward direction vector of the palm, \vec{u} is the normalized up direction vector of the palm, and \vec{r} is the normalized right direction vector of the palm, as depicted in Figure 3.13.

This hand tracking approach enables Robody's hands to directly respond to and track the movements of the user's hands, resulting in a strong sense of embodiment.

Mapping the position of the Robody's arms in the user's field of view in the AR environment to the real arm position of the user can provide a better sense of embodiment. According to Hapuarachchi et al., while asynchronized arm positions can still induce a sense of body ownership, synchronized arm positions can significantly enhance the sense of embodiment [134].

So far, due to the difficulty of accessing Magic Leap 1's raw camera stream, it is significantly more practical to use a webcam for development and debugging purposes. During testing, to better simulate the camera of an AR HMD, the webcam needs to be mounted on the user's head. This results in a significant offset between the webcam's field of view and the user's

field of view, making it challenging to accurately map the position of the Robody's arms in the user's field of view in the AR environment to the real arm position of the user. It's important to note that in future developments, this webcam setup, including Magic Leap 1, will be replaced for better performance.

The hand tracking control method is illustrated in Figure 3.14.

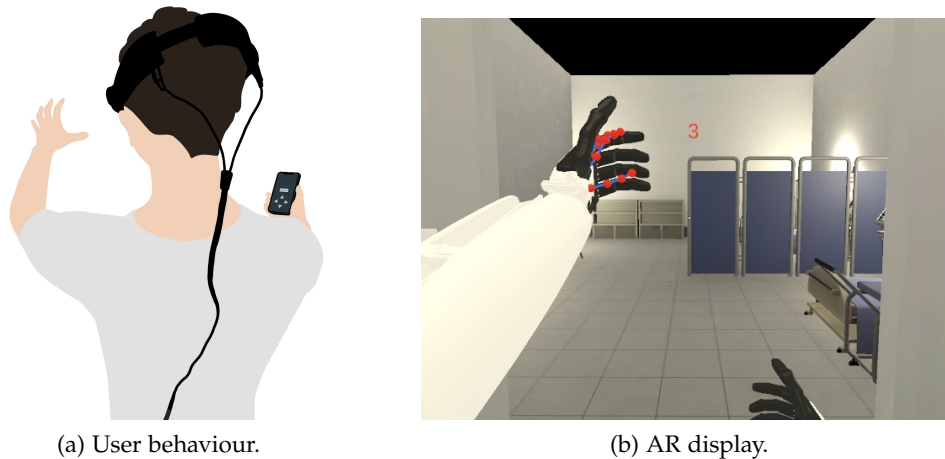


Figure 3.14.: Illustration of the hand tracking control mode.

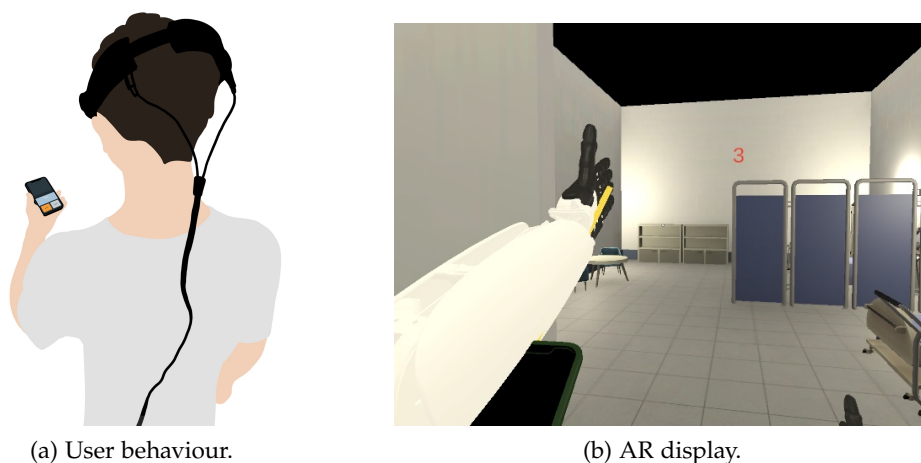
Smartphone "Pointer" Implication

In the previous discussion on Robody hand control strategies, we mentioned the limitations of smartphone motion sensors. Modern smartphones are typically equipped with micro-electro-mechanical systems (MEMS) sensors, including three-axis accelerometers, three-axis gyroscopes, and magnetometers for motion and pose detection. Among these, the accelerometer can be used to detect linear acceleration in three-dimensional space, and by integrating this detection twice, the translation of the device in three-dimensional space can be obtained. However, due to the insufficient precision of smartphone accelerometers, internal friction, the influence of gravity, and the amplified error caused by double integration, the translation information detected and deduced by the accelerometer contains a considerable amount of error. Additionally, gyroscopes and magnetometers are unable to measure linear acceleration, velocity, or translation, thus they are not capable of correcting the data produced by the accelerometer. Therefore, relying only on the smartphone accelerometer cannot provide effective support for measuring user hand translation.



Figure 3.15.: Hand tracking control mode concept.

However, in contrast, smartphones are capable of measuring the device's orientation with a fairly good accuracy. The three-axis gyroscope can detect the rotational rate of the device in three dimensions and obtain changes in the device's orientation through integration. Since the measurement data from the gyroscope has smaller errors compared to accelerometer data [48], and both the accelerometer and magnetometer can correct the device's orientation, in practice, we can rely on the orientation information provided by the device's motion sensors and use it as support for controlling Robody.



(a) User behaviour.

(b) AR display.

Figure 3.16.: Illustration of the smartphone pointer control mode.

3. Approach

In subsection 2.4.2, we discussed various cases where only smartphone orientation data is utilized for control. Taking into account the situations described above, and referring to these cases, we abandoned the use of linear acceleration data provided by the accelerometer and designed a Robody hand control method that relies only on the device's orientation data. This approach was inspired by the Magic Leap 1 controller. The Magic Leap 1 controller casts a ray from its front end in the augmented environment, allowing users to point into the desired direction with the controller and select the objects with which the ray intersects. In this context, the controller is referred to as a "pointer." Our design was inspired by this "pointer" idea: we placed a virtual smartphone at a fixed location near the user's hand position in the AR environment, and the orientation information measured by the user's handheld real smartphone was transmitted to the AR environment, so that the orientation of the virtual smartphone could match that of the real one. The virtual smartphone emits a ray from its front end, and the length of the ray (the distance between the hand position and the user position) is controlled by the user sliding up and down on a fixed area of the smartphone screen. We mapped the position of Robody's wrist to the end of the virtual ray, allowing it to move in the direction pointed by the smartphone. Users can perform specific hand actions of Robody, such as grasping, using buttons on the smartphone screen. In this control method, although Robody's wrist can no longer rotate freely, we can achieve free movement in three-dimensional space for Robody's hands relying only on the orientation information provided by the smartphone's motion sensors. The concept is depicted in Figure 3.15.

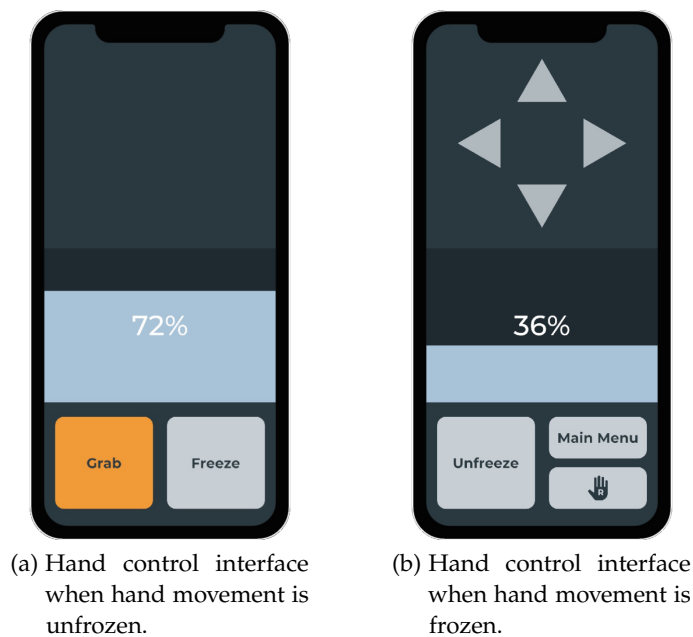


Figure 3.17.: Hand control interface (Smartphone Pointer Mode).

The grab button and slide control are designed within easy reach of the thumb of the hand holding the smartphone, allowing the user to interact with the application with one hand.

When the user switches between left and right-hand control, the button layout also changes accordingly to facilitate one-handed utilisation for the user.

However, this control method still has many limitations. Firstly, due to the lack of translation tracking, the positions of the virtual and real smartphones cannot correspond to each other precisely, which may affect the accuracy of pointing. Secondly, the user's hand movements do not match Robody's hand movements, which can negatively affect the user's sense of embodiment. Additionally, as mentioned earlier, it is difficult to intuitively control Robody's finger movements through smartphones. Furthermore, since there is only one smartphone, the user can only control one of Robody's arms at a time. Finally, because this control method is not straightforward, users need to invest a considerable amount of time and effort to learn and adapt to it in order to proficiently complete tasks.

The smartphone pointer control method is illustrated in Figure 3.16.

Smartphone 6-DoF motion tracking

The aforementioned smartphone control solution represents a compromise made when the smartphone's accelerometer lacks sufficient accuracy. According to Hapuarachchi's experimental results [134], such a control method, due to the mismatch between Robody's hand movements and the user's hand movements, may result in a poorer sense of embodiment. Additionally, due to this indirect control method, users may require more learning and practice to improve control experience and efficiency.



Figure 3.18.: Hand tracking control mode concept.

Inspired by Babic et al. [53], we see the potential of using the smartphone camera as an additional sensor within the existing AR solution framework, along with motion sensors,

to provide 6 DoF motion tracking for the smartphone. Their work is highly relevant to our situation. Babic et al. successfully implemented 6 DoF motion tracking on the iPhone using ARKit, which was introduced in subsection 3.2.2. Similarly, based on iPhone development, we also chose ARKit, to provide 6 DoF device tracking for our control scheme. ARKit utilizes the device's built-in camera and motion sensors to predict and calibrate the smartphone's orientation and motion state using SLAM algorithms and sensor fusion. The concept is depicted in Figure 3.18.

Tracking the user's hand pose with a smartphone in 6 DoF allows the virtual Robody's hand to move in sync with the user's hand movements. The wrist of Robody can also rotate based on the smartphone's orientation, providing more control freedom than the previously mentioned pointer method. This control method is similar to the approach implemented by Kawabata and Gao in their works using the Magic Leap 1 controller, where the hand position of Robody in the AR environment is directly mapped to the user's actual hand position, aiming for maximum visual overlap. This also means that users do not need to expend excessive effort to familiarize themselves with the control of the robot's hands.

Similar to the smartphone pointer control method described above, Robody's hand movements (such as grabbing) are achieved through buttons on the touchscreen. With the same one-handed operability, the UI layout also remains consistent with previous control method, with the exception that the slider control is removed.

The smartphone motion tracking control method is illustrated in Figure 3.19.

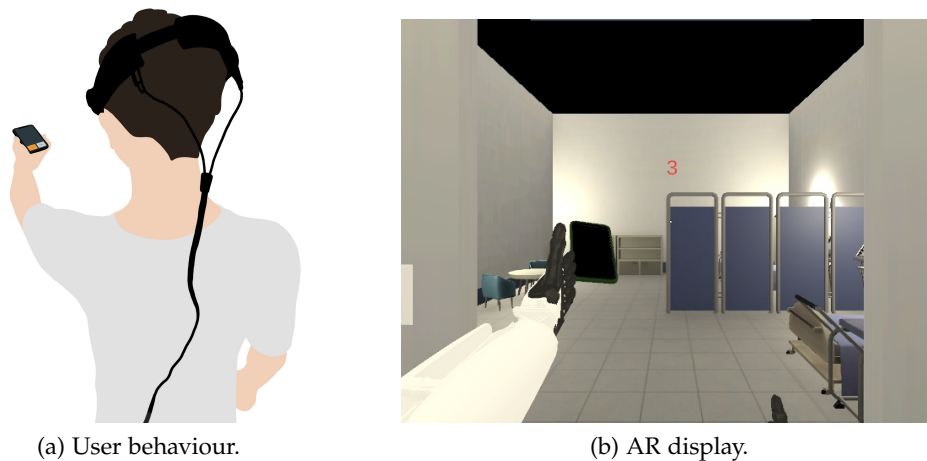


Figure 3.19.: Illustration of the smartphone motion tracking control mode.

3.3.8. Finger Control

Smartphone-Based Control

Both the smartphone and controller can effectively control Robody's hand position and orientation. However, they lack effective methods for controlling finger movements, as Gao

pointed out in his work [149]. Like healthy human beings, Robody, as a musculoskeletal robot, has ten fingers, each with three joints controlled by biomimetic tendons to bend and stretch the fingers. However, controllers typically have only a few buttons available for use (as shown in Figure 3.4), usually ranging from 2 to 4 buttons, making it impractical to control all the fingers individually at the same time.

Smartphones face similar issues. While the screen can provide any number of buttons, sufficient to control each finger individually, in practice, the hand holding the phone can only interact with the screen using the thumb. If the user needs to interact with more than one button simultaneously, they often need to use their other hand to manipulate the screen. This means that if simultaneous control of multiple fingers is required, the user must use their other hand.

The hand position and orientation of the virtual Robody is deduced from the hand holding the smartphone (e.g., the left hand). However, if the other hand (e.g., the right hand) needs to be involved (e.g., moving the left hand holding the phone to control Robody's hand position and orientation, while the right hand taps on the phone screen to control finger opening and closing), it can disrupt the sense of embodiment. First, the user needs to map the screen taps to finger bending and stretching. Second, the user needs to map the movements of one hand to the other hand of Robody (e.g., using the right hand's finger taps to control Robody's left hand's fingers), which can be challenging and require learning.

Considering the above, our solution is to use only one button on the smartphone screen to control Robody's fingers. This is similar to the approach used by Kawabata and Gao in their works, as well as the existing VR control method developed by Devanthro, which use a single controller button to control finger movements to complete the grabbing action (although the work of Kawabata and Gao did not visualize virtual Robody's finger movements, their implementation of the grabbing action is fundamentally controlling the fingers).

As mentioned in subsection 3.3.7 and subsection 3.3.7, we positioned the grab control button near the thumb of the phone-holding hand for convenient one-handed operation. Specifically, while we used the same button layout in both the smartphone pointer and smartphone 6-DoF motion tracking control methods, the logical design of the button differs.

However, in practice, as the users have to adjust their hand pose continuously and need to occasionally interact with the slider using the thumb, continuously holding the grab control button is not practical (e.g., when the users need to change the distance between the hand and the user while holding an object in smartphone pointer control mode). Therefore, our design allows users to tap the grab button to make Robody grab and hold until the grab button is tapped again, at which point Robody releases its grip. This allows users to easily switch between grabbing and controlling the hand pose and the hand-to-head distance, with a cost of a slight compromise in the sense of embodiment.

Hand Tracking-Based Control

Hand tracking can solve the issue of full finger control. The hand landmarks recognition feature provided by MediaPipe can detect the real-time 2.5D positions of 21 hand landmarks [117] (as shown in Figure 3.12), which allows us to further compute the pose of the connecting

bones between these landmarks. Furthermore, utilizing the pose information of adjacent connecting bones, we can compute the relative angles of the joints.

Lee et al. introduced a 27-DoF hand model [150] (as shown in Figure 3.20). In the rest of the thesis, we will use the joint naming conventions from this model for clarification.

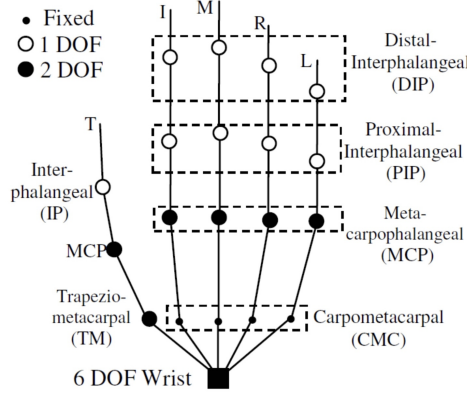


Figure 3.20.: Hand model introduced by Lee et al. [150].

Although it is possible to individually track the angle of each joint, independently controlling the rotation of each joint is not necessary. Few people can independently bend their PIP and DIP joints of their fingers. According to the biomechanical studies by Kuch et al. [151], certain closed-form constraints can be derived, and one of the most important is:

$$\text{PIP} = \frac{3}{2} \text{DIP}$$

To reduce real-time computational load and mitigate the inconsistency introduced by unstable hand tracking, we adopted a strategy similar to Gump et al., limiting the degrees of freedom to 1 DoF for flexing in each finger. This is sufficient to control the hand to perform most gestures. We chose the angle of the PIP joint as the degree of finger flexion. We predefined the PIP joint angles for fully stretching and fully bending for each finger. The final posture of the fingers is obtained by interpolating between these values using the current computed PIP joint angle. The interpolated value is then used to define the degree of flexion.

The PIP joint angle of a single finger is computed using 3 detected landmarks:

$$\begin{aligned} \vec{v}_1 &= l_1 - l_2 \\ \vec{v}_2 &= l_3 - l_2 \\ \theta &= \cos^{-1} \frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1| \cdot |\vec{v}_2|} \end{aligned}$$

where l_1, l_2, l_3 are positions of adjacent landmarks of a finger, representing MCP, PIP, DIP joints respectively, and θ is the angle between \vec{v}_1 and \vec{v}_2 , as depicted in Figure 3.21.

While tracking the bending degree of the fingers individually is theoretically sufficient for the physical Robody to perform any hand gesture, in the virtual environment, we still need

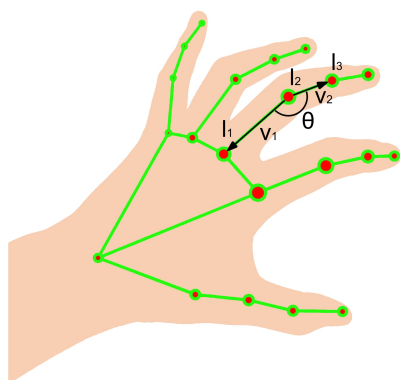


Figure 3.21.: Demonstration of finger joint angle calculation.

to determine whether a gesture is a "grabbing" gesture for interaction with virtual objects. MediaPipe also provides gesture recognition capabilities and can recognize a closed-fist gesture. However, in practice, the grabbing gesture is not a standard closed-fist gesture, making it difficult for MediaPipe to recognize. Furthermore, additional recognition would significantly increase the computational load on the hardware, affecting the program's real-time response speed. Therefore, we aim to directly use the calculated joint angle data to determine the grabbing gesture.

Due to self-occlusion of the hand and the instability of hand tracking, our final solution is to consider the user's gesture as "grabbing" when at least four fingers exceed a predefined bending threshold. This allows interaction with virtual objects in the environment that can be grabbed.

3.3.9. Displacement

In practical situations, precise adjustment of Robody's position is often required. As mentioned earlier, we introduced a task assignment system that allows Robody to autonomously perform long-distance displacements. However, this task assignment system is not suitable for precise position adjustments. In previous works by Kawabata and Gao, Robody's position was controlled based on the displacement of the AR HMD, meaning that any movement of the user would cause Robody in the AR environment to move accordingly. However, in real clinical environments, making Robody precisely follow the user's every move is unsafe and impractical. Since clinical settings may have various potential hazards, such as narrow spaces, the presence of other medical equipment, uncertainty in the movements of healthcare personnel and patients, and more. If Robody were to blindly follow these movements, it could lead to collisions, accidents, or unnecessary disruptions. Furthermore, the physical space where the user is situated might not be suitable for large-range movements.

Gao proposed a button-based Robody locomotion mode within the AR environment. Users

would use the AR controller to click on directional buttons, triggering or stopping Robody's movement in the respective direction. However, due to the need to aim at virtual buttons, using AR controllers may not allow for quick instruction delivery to Robody or timely responses.

Inspired by Gao's button-based locomotion mode, we took a different approach. Instead of embedding directional buttons within the AR environment, we integrated them into the smartphone application, as shown in Figure 3.3. This means that users no longer need to aim at floating virtual buttons in the AR environment; they can simply tap on the smartphone screen to control Robody. This significantly improves response speed and operational efficiency. Since the virtual Robody moves on wheels, these buttons allow users to control its forward and backward movements, as well as left and right rotations. The layout of the buttons is also inspired by common household appliance remote controls, making it easy for users to quickly learn and proficiently use these functions.

4. Evaluation

4.1. Introduction

4.1.1. Overview

In this chapter, we conducted a comprehensive assessment of the performance of our developed Robody control system which consists of various control strategies. Our experiment is divided into two parts: the first part involves integrating our implementation with the physical Robody and testing whether our method can achieve the desired outcomes. The second part of our experiment includes comparing the performance of the different control methods. Throughout the study, we collected feedback and data from participants, identifying areas for improvement, and evaluating user perceptions of system performance. The results of the evaluation helped determine which control strategy performs the best or is most suitable for specific tasks and applications.

4.1.2. Goals

To effectively evaluate the project, well-defined goals and expected outcomes are essential. These goals will help us validate the system's performance and usability, while also guiding our experimental designs during the evaluation process. The following are the evaluation goals we have established:

1. **Successful integration:** Working with Devanthro to deploy our implemented control method on the Physical Robody and evaluate its usability and performance.
2. **Performance validation:** Validating the performance of various Robody control strategies. We will compare the accuracy, stability and complete time of user task execution using different control methods.
3. **System usability:** We will evaluate the usability of the control system we have developed, including the system's stability, complexity, and user-friendliness.
4. **Embodiment:** We will evaluate the degree of embodiment that our developed control system provides to users during operation.
5. **User feedback analysis:** Gathering user feedback and suggestions for future improvements. This feedback will provide us with valuable insights and serve as a crucial resource for ameliorating our implementation in the future.

4.2. Experimental Setup

4.2.1. Integration with Physical Robody

Robody Simulation in ROS

The first part of the evaluation involves integrating our implementation with the physical Robody. However, as mentioned in subsection 2.2.1, to avoid unnecessary debugging and time wastage, it is essential to validate the feasibility of our approach in the simulation before conducting tests on the physical Robody. Our initial step of the first experiment is to simulate upper limb movements in Robody using the ROS interface provided by Devanthro.

To perform the simulation, we contacted Devanthro and obtained the simulation software for Robody ¹. This simulation software is a ROS bridge Docker image. The interface of this simulation software is identical to that of the physical Robody. If we can successfully control the simulated Robody using our implementation, it theoretically means that our implementation can also control the physical Robody.

The workflow for the simulation is as follows:

1. Run the Robody simulation image within a Docker container.
2. Install and run the ROS-Unity TCP Endpoint package within the container.
3. Install the ROS-Unity Connector in Unity and publish relevant joint data to the specified topic.
4. The Robody simulation receives the published joint data and performs the corresponding movements.

The software infrastructure is depicted in Figure 4.1.

After successfully establishing communication between the simulation program and Unity, we can make the Robody in the ROS simulation follow the movements of the virtual Robody in Unity.

However, it is worth noting that the upper limb mobility of the Robody in the ROS simulation is limited by the physical capabilities of the physical Robody. For example, the upper arm of the Robody in the ROS simulation can only be raised to an angle of about 60 degrees. On the contrary, in Unity, when using Unity's native humanoid IK to control the virtual Robody's arms, the range of motion is biomechanically accurate, allowing the upper arms to be easily raised overhead. This difference implies that the Robody in the ROS simulation cannot fully imitate the movements of the virtual Robody in Unity. To address this issue, we abandoned the use of Unity's native humanoid IK to control the Robody's arms and instead re-implemented an IK system. This new IK system controls the joints of the virtual Robody in Unity in the same way as the ROS simulation and apply constraints on joint mobility angles to fully replicate the Robody's mobility in the ROS simulation. With the help of the new IK system, the joint rotation data of the virtual Robody in Unity can

¹cardsflow_docker. https://github.com/CARDSflow/cardsflow_docker

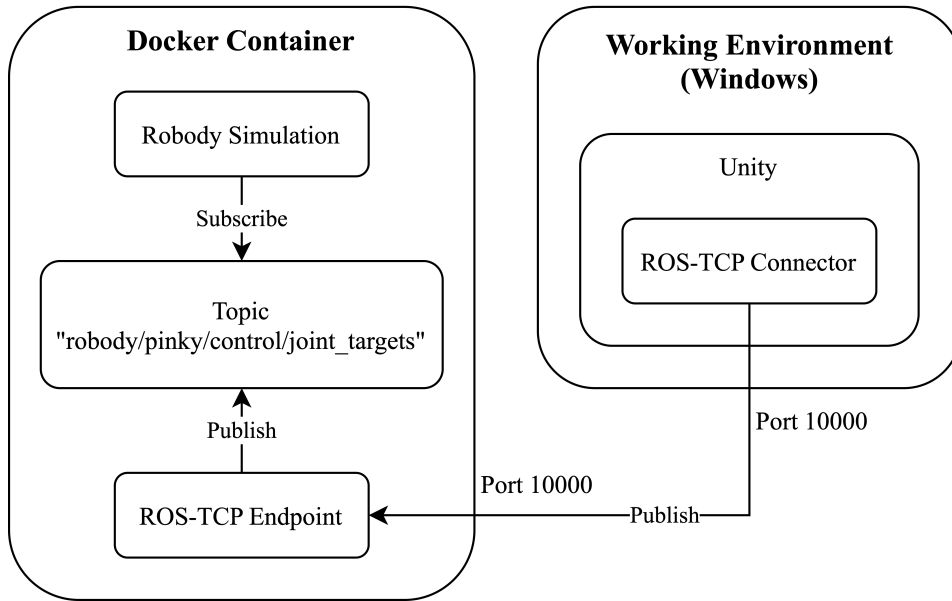


Figure 4.1.: Software setup of the ROS simulation.

correspond perfectly with the Robody in the ROS simulation, allowing both to perform the same actions (as shown in Figure 4.2). However, the drawback of this approach is that the Robody's hands cannot fully follow the user's movements, as sometimes they go beyond the constrained range.

Test on Physical Robody

After successfully controlling Robody within the simulation program, we have all the prerequisites needed to test our control method on the physical Robody. With the assistance of Devanthro, we utilized the existing infrastructure of Robody, by making some adjustments, such as connecting to the same local network where the physical Robody is connected to and modifying the target topic for publishing joint data, we successfully enabled our control method to operate the arm of the physical Robody, as shown in Figure 4.3.

4.2.2. Tasks in the Virtual Environment

In addition to testing our control system on the physical Robody in the first experiment, the second experiment consists of various small tasks that can be accomplished within the virtual clinical environment. In this experiment, we aim to compare the efficiency and usability of different control modes.

According to our concept, users will need to complete the following small tasks within the virtual clinical environment simulated in their AR HMD:

1. **Task 1:** Move Robody to a specified location. The users have the option to either

4. Evaluation

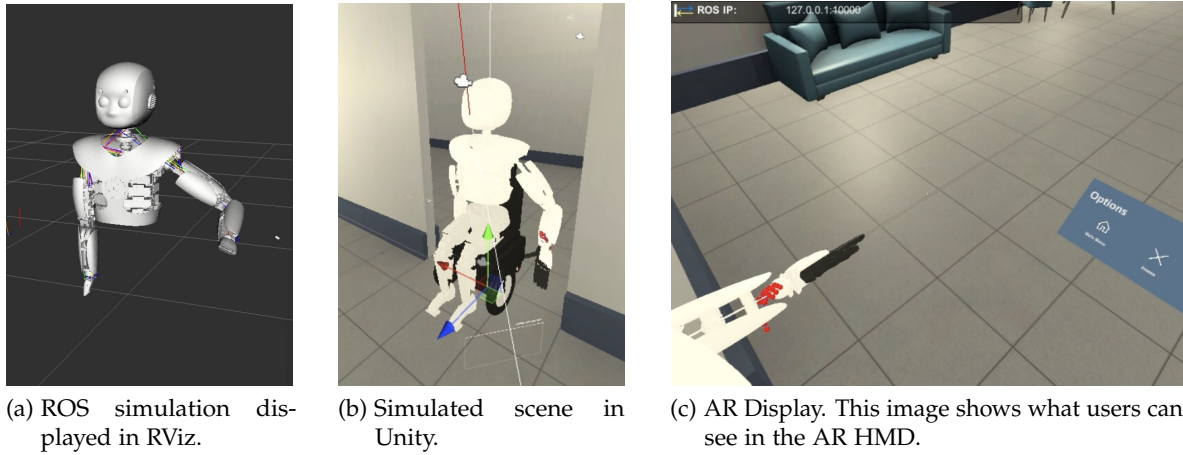


Figure 4.2.: Demonstration of ROS simulation integration.

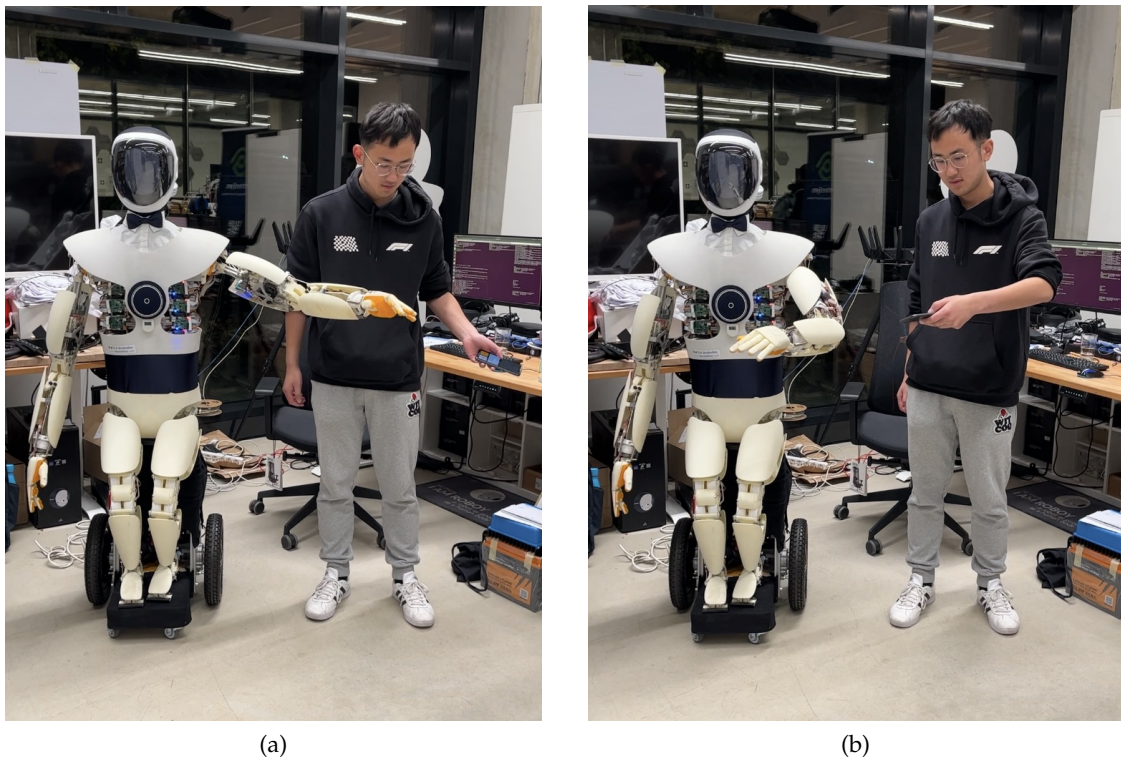


Figure 4.3.: Successful integration with physical Robody.

use autonomous task assignment to displace Robody, manually steer Robody to the specified location, or a combination of both.

2. **Task 2:** Use Robody to turn off the lights in a designated room. Users should first move Robody to the light switch in front of the designated room. Then, use arm control mode to trigger the light switch.
3. **Task 3:** Use Robody to move a bottle of water from one specified location to another. Users should first control Robody to reach the storage room, where they can use arm control mode to grab a bottle of water. After that, they should bring it to the specified location.

The tasks are not randomly designed; each task has a specific focus. The first task is designed to assess user performance when combining autonomous task assignment and manual steering of Robody. The second task evaluates user accuracy in controlling the arm and hand-to-object distance. The third task simulates a scenario where users need to "deliver water to a patient", to comprehensively assess their performance in steering Robody, controlling the arm, and grabbing objects.

4.3. User Study

4.3.1. Participant Consent

Before participating in our experiment, we required all participants to provide their consent. We prepared a comprehensive consent form that includes a description of the experiment, the estimated time required for participation, the data involved in the experiment, the risks of participating, payment details, and participant's rights. At the end of the form, we provided our contact information for participants to reach out if they had any questions. The consent form is provided in section A.1.

4.3.2. Experiment

Let's revisit the three tasks designed in subsection 4.2.2:

1. **Task 1:** Move Robody to a specified location.
2. **Task 2:** Use Robody to turn off the lights in a designated room.
3. **Task 3:** Use Robody to move a bottle of water from one specified location to another.

Phase 1

Participants, after the introduction to the system, are guided through the various functions and control methods of our system. During this process, users can gain a basic understanding and become familiar with the entire system.

Phase 2

All participants are required to start with the task 1 as it is independent of hand control methods, involving only smartphone-based operations. The time taken by each participant to complete this task is recorded for further analysis.

Phase 3

In the subsequent experiment, we categorized the hand control methods into two groups: smartphone-based control methods and the hand tracking-based control method. Also, all participants were divided into two groups as follows:

- Group A: Starting with the smartphone-based control method, followed by the hand tracking-based control method.
- Group B: Starting with the hand tracking-based control method, followed by the smartphone-based control method.

After participants have completed task 2 and 3 using the first control method, Both groups switch the control mode they are using and complete the same two tasks again.

The time taken for each task was recorded for subsequent analysis. While participants were encouraged to complete all experiments, they were free to choose to abandon a given task. Unfinished task data was not included in the analysis.

Phase 4

Finally, participants provided feedback on their experiences by completing two separate questionnaires. Participants were also encouraged to share any suggestions and comments. The experiment flow is depicted in Figure 4.4.

4.3.3. Questionnaire

Our experiment not only covers the usability aspects of our implemented system but also focuses on users' sense of embodiment during usage. Therefore, we have prepared three questionnaires, two standardized questionnaires and one specific to our control system. These questionnaires collect participants' feedback, helping us gain a comprehensive understanding of system usability and user embodiment of our implementation through analysis.

System Usability Scale (SUS)

In order to evaluate users' feeling about the usability our implementation, we employed the System Usability Scale (SUS) proposed by Brooke as the base of our first questionnaire [152]. SUS is a reliable, low-cost usability scale that can be used for global assessments of system usability. It covers various aspects of system usability, including the need for support, training, and complexity. The questionnaire is provided in section A.2.

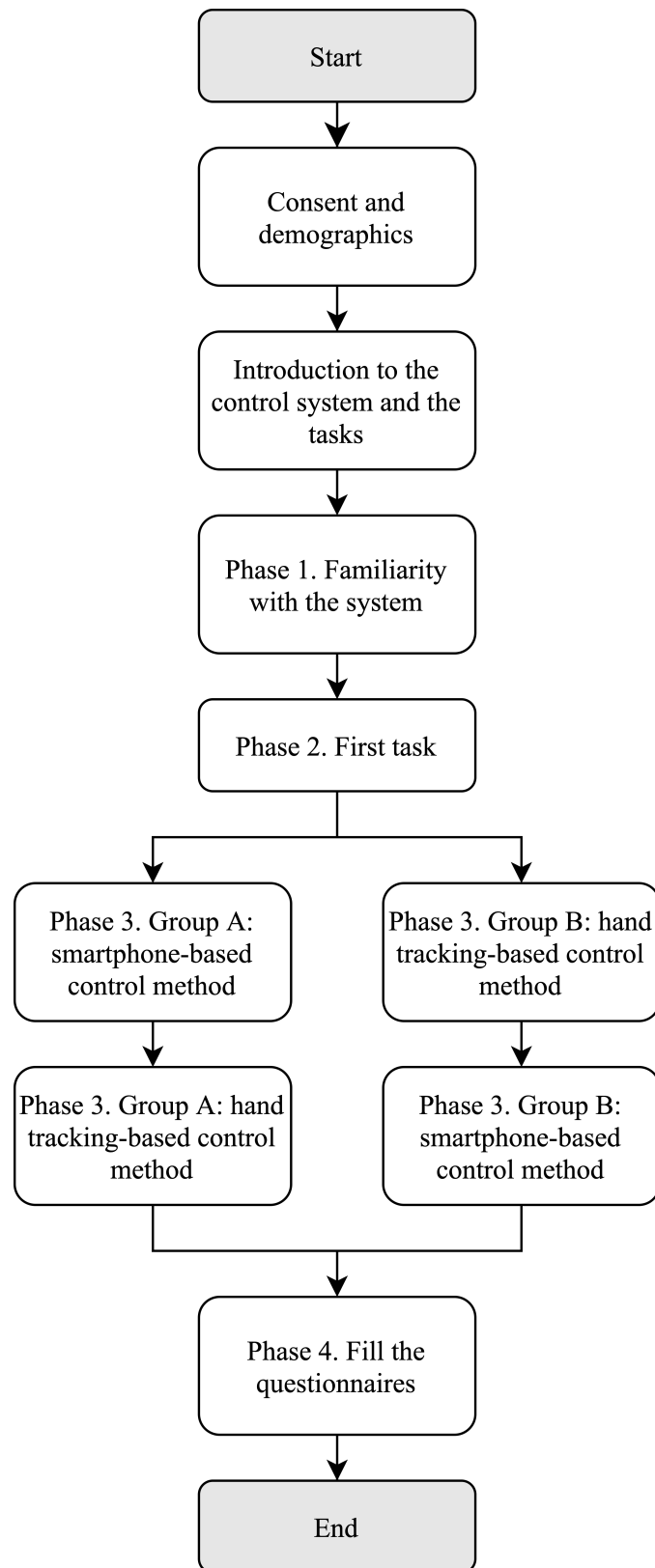


Figure 4.4.: Experiment flow.

The SUS consists of 10 questions related to users' subjective perceptions of system usability. It uses a 5-point Likert scale. We applied the score computation method also proposed by Brooke simultaneously:

"To calculate the SUS score, first sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For items 1,3,5,7,and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of SU. SUS scores have a range of 0 to 100."

Embodiment Questionnaire

For the standardized measurement of users' sense of embodiment, we have chosen the standardized questionnaire on avatar embodiment proposed by Peck and Gonzalez-Franco [153]. This questionnaire has been widely applied in the studies of virtual entities and role-playing experiences and has been validated as an effective measurement tool [134, 154].

The original questionnaire consists of 16 questions that focus on user interactions and sensations regarding their body's interaction with the virtual body. The questionnaire uses 7-point Likert-scale ranging from strongly disagree to strongly agree. Peck and Gonzalez-Franco also introduced a score computation method for this questionnaire, which will further assist us in quantifying users' sense of embodiment of the virtual Robody more accurately. By employing this standardized questionnaire, we can better compare users' embodiment experiences and make our study results more easily comparable to other relevant studies. The questionnaire is provided in section A.3.

The score computation proposed by Peck and Gonzalez-Franco is as follows [153]:

$$\begin{aligned}\text{Appearance} &= (R_1 + R_2 + R_3 + R_4 + R_5 + R_6 + R_9 + R_{16})/8 \\ \text{Response} &= (R_4 + R_6 + R_7 + R_8 + R_9 + R_{15})/6 \\ \text{Ownership} &= (R_5 + R_{10} + R_{11} + R_{12} + R_{13} + R_{14})/6 \\ \text{Multi-Sensory} &= (R_3 + R_{12} + R_{13} + R_{14} + R_{15} + R_{16})/6 \\ \text{Embodiment} &= (\text{Appearance} + \text{Response} + \text{Ownership} + \text{Multi-Sensory})/4\end{aligned}$$

where R_i represents the result of question number i . The final embodiment score will be in a range from 1–7 indicating low to high embodiment.

Project Specific Questionnaire

At the end, we end with a short questionnaire consisting of four questions. In this questionnaire, we briefly asked users about participants' prior experience using AR/VR, their preferred control methods, their opinions on some known issues and their suggestions for the system. Only two multiple choice questions in this questionnaire are mandatory to reduce the burden on participants as much as possible. The questionnaire is provided in section A.4.

4.4. Results

4.4.1. Integration with Physical Robody

In the first part of the evaluation, with the assistance of Devanthro, we successfully utilized our control method to control physical Robody. This marks a significant step toward the practical application of this control system.

We are delighted to report that many aspects of integration met or exceeded our expectations:

- Robody's arms were successfully controlled by our program, indicating that our IK-based arm control implementation is correct and scalable. By moving the IK target in the virtual AR environment, we could change the virtual Robody's arm pose and, subsequently, change the physical Robody's arm pose.
- Robody's arms could closely follow the user's control actions in real-time. With the local area network connection in the laboratory, the delay was minimal, barely noticeable to users.
- Robody's pose closely matched that of its digital twin created in Unity, demonstrating good consistency.

However, we also identified some issues for improvement:

- Physical Robody's arm movements were not as smooth as desired. When users directed the IK target at certain angles or positions, Robody's hand movements became uneven. This issue may be caused by the suboptimal performance of our IK solver. Utilizing a more robust and bionic IK solver is expected to mitigate this problem.
- Robody, being a robot, its joints have fewer degrees of freedom and less flexibility in its range of motion compared to humans. In situations where the joints' movement is restricted, Robody's hands may not reach the IK target as expected. This discrepancy can lead to inconsistencies between user and Robody movements, reducing the sense of embodiment.

Overall, the integration with physical Robody was successful, and had fulfilled our established goals.

4.4.2. User Study Demographics

Our user study involved a total of 7 participants, one of whom is a member of the Robody student team during the current semester (Winter Semester 2023). All participants were students from TUM with various majors, including computer science, data engineering, electrical engineering and aerospace.

All participants had normal or corrected-to-normal vision and did not report any physical impairments that would affect their ability to interact with the system.

According to the participants' reports, more than half of them are relatively unfamiliar with AR/VR devices (Never/Rare). One participant considered himself very familiar with AR/VR.

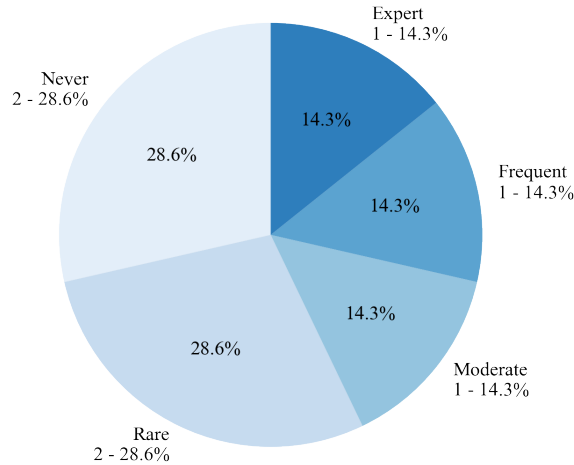


Figure 4.5.: Pie chart: prior experience with AR/VR.

4.4.3. System Usability

For each participant's response to the first questionnaire, we used the score computation method proposed by Brooke et al. to calculate the system usability score for our control system.

According to Sauro's summary of over 500 SUS surveys, a SUS score above 68 is considered above average, while a score below 68 is below average ².

Based on our final results, the average system usability score for our implementation is 70.71 (out of 100, rounded to two decimal places), with a median score of 72.5. The highest score is 80, and the lowest score is 52.5. This average score is slightly above the average of 68. The results of the investigation are presented in Figure 4.6 in the form of a box plot. The box plot also indicates that our data is positively skewed, suggesting that the majority of participants rated the usability of the system relatively high.

It's worth noting that, according to Sauro's investigation ³, difficult tasks can reduce SUS scores by an average of 8%, and for users with less experience, this reduction can even reach close to 20%. In our survey, most participants were relatively unfamiliar with AR/VR, and they had insufficient exposure to our implementation beyond the introduction session and initial attempts before conducting the tasks. Additionally, our experiment involved some challenging tasks with complex logic (e.g., users need to freeze Robody's hand movements

²Measuring Usability with the System Usability Scale (SUS). <https://measuringu.com/sus/>

³SUSTisfied? Little-Known System Usability Scale Facts. <http://uxpamagazine.org/sustified/>

before moving or returning to the main menu, and having to unfreeze Robody’s hand movements again after re-entering hand control mode to activate hand control). Taking these factors into account, our SUS score may have been negatively influenced. SUS is also considered to measure the learnability of the system. Some participants reported that they need more time to learn and become familiar with the system for optimal use.

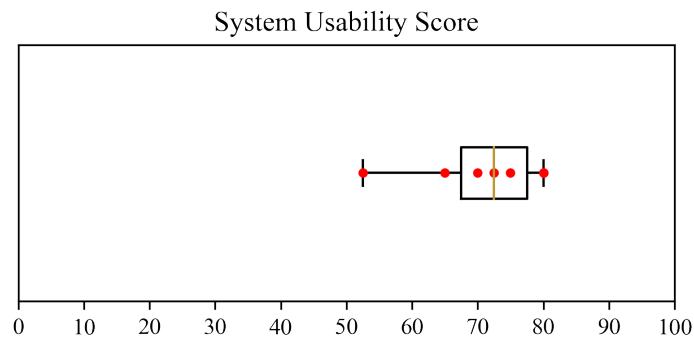


Figure 4.6.: Box plot: system usability score. Red dots represent original data; the orange line represents the median.

4.4.4. Embodiment

In the standardized questionnaire proposed by Peck et al. [153], embodiment is divided into four sub-scales: Appearance, Response, Ownership, and Multi-Sensory. We used the score computation method mentioned in subsection 4.3.3 to compute each participant’s scores for the four sub-scales and the final embodiment score.

According to our final results, the average scores for Appearance, Response, Ownership, and Multi-Sensory are 4.48, 4.02, 4.36, 4.05 (on a scale of 1-7, all rounded to two decimal places), with median scores of 4.75, 4.00, 4.83, 4.33 (all rounded to two decimal places).

The average scores for these four sub-scales are all slightly above the midpoint of 4, with little variation. The median values show a greater difference, with the Response score noticeably lower. This may reflect that participants felt fewer stimuli to their own bodies during the experiment, possibly due to a lack of sound and haptic feedback. The results of the investigation for the four sub-scales are presented in box plots in Figure 4.7. In the box plots, we observe that the scores for Appearance are concentrated and relatively high, indicating that most participants can perceive embodiment from the visual appearance of Robody. Ownership and Multi-sensory show a slight positive skew, suggesting that more participants feel ownership of a part of Robody’s body and can perceive Robody’s sensations in the virtual environment [155].

Based on the computed scores of the sub-scales, the average embodiment score for our implementation is 4.23 (on a scale of 1-7, rounded to two decimal places), with a median score of 4.51 (rounded to two decimal places). The highest score is 5.25, and the lowest score is 2.56 (rounded to two decimal places). The results of the investigation are presented in Figure 4.8 in

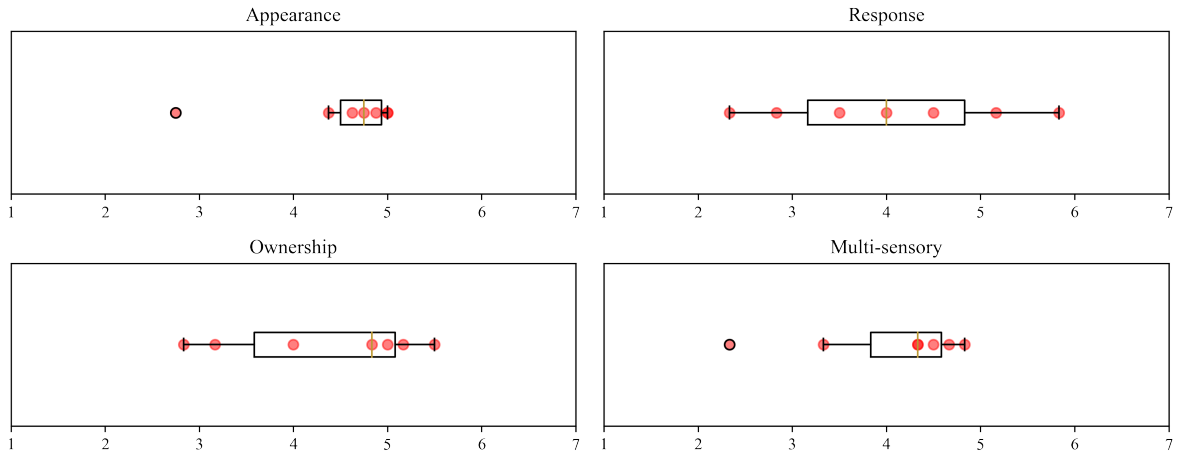


Figure 4.7.: Box plot: sub-scales of embodiment.

the form of a box plot. From the box plot, we can observe a positively skewed distribution of the data. With over half of the scores exceeding 4 points, it can be considered that our system is capable of inducing a certain level of embodiment. However, it is noteworthy that while a substantial portion of participants scored above 4 points, the scores are not exceptionally high. Additionally, a significant number of participants gave scores below 4 points, suggesting notable shortcomings in our system’s ability to induce a strong sense of embodiment.

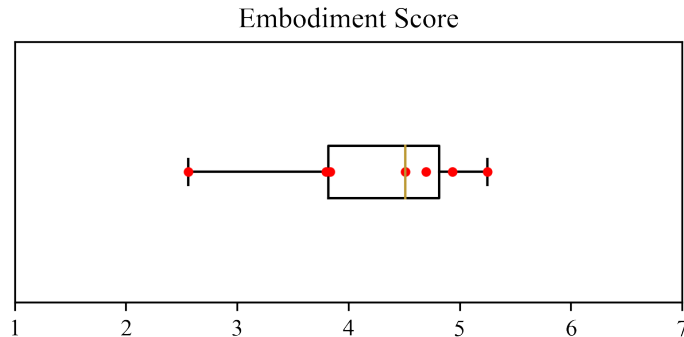


Figure 4.8.: Box plot: embodiment score. Red dots represent original data; the orange line represents the median.

We additionally explored the relationship between participants’ system usability scores and embodiment scores. We presented each participant’s system usability score and embodiment score respectively as x and y-axis coordinates, through a scatter plot in Figure 4.9. Two clear outliers can be observed: (52.50, 3.80), (75.00, 2.56) (rounded to two decimal places). These two experiment participants gave notably low scores for either system usability or embodiment. We performed two linear regressions on the data, one excluding these two outliers and one including all data points. As shown in Figure 4.9, excluding these outliers for linear

regression reveals a pronounced trend: there is a noticeable inverse relationship between usability scores and embodiment scores. That is, higher embodiment scores correspond to lower usability scores. This finding was unexpected, and we speculate that a heightened sense of embodiment might upgrade users' perception of the virtual environment, and potentially degrades users' perception of the real-world environment, leading to extra difficulties in hardware operation (e.g., using a smartphone). When considering all data points for linear regression, a nearly horizontal line is obtained, suggesting no apparent correlation between system usability scores and embodiment scores.

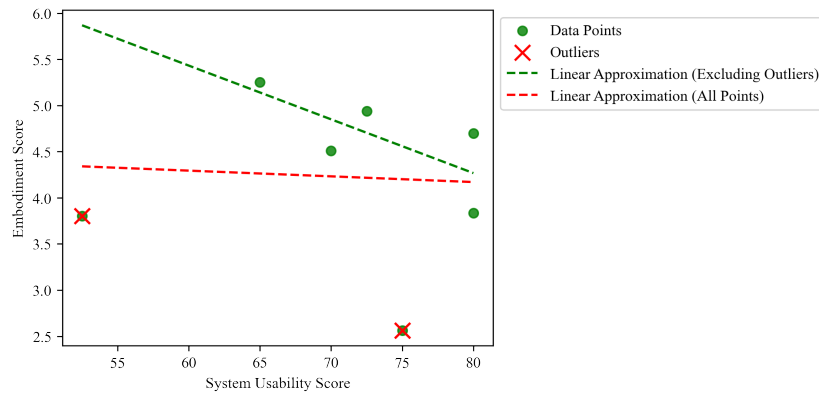


Figure 4.9.: Scatter plot: correlation between system usability scores and embodiment scores.

4.4.5. Project Specific Issues

Preferred Control

We investigated users' most preferred and least preferred control methods. The results of the investigation are presented in Figure 4.10 in the form of a bar chart.

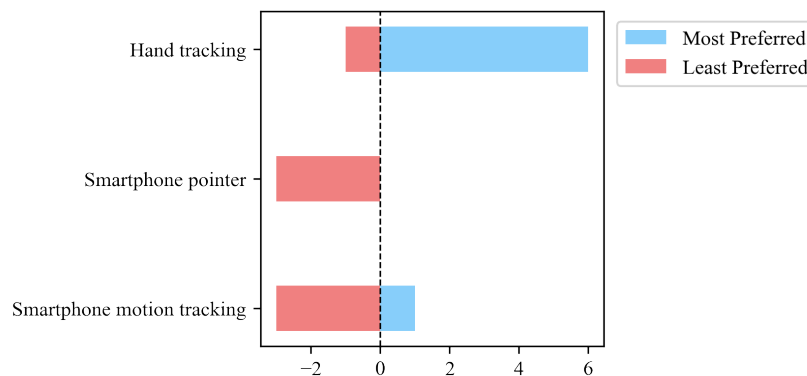


Figure 4.10.: Bar chart: Which control methods do you like the most and the least?

The majority of users chose hand tracking as their favorite control method. Some partici-

pants mentioned that hand tracking-based control is the most intuitive, requiring minimal learning curve. On the other hand, smartphone-based control methods, especially the smartphone pointer control, along with the need for frequent switching between frozen/unfrozen states, made it challenging for some participants to learn quickly. This aligns well with our expectations. Smartphone motion tracking, while still functional, was less favored due to occasional instability issues, such as positional drift.

Known Issues

We also investigated users' perceptions of some known issues in the control system. The results of the investigation are presented in Figure 4.11 in the form of box plots.

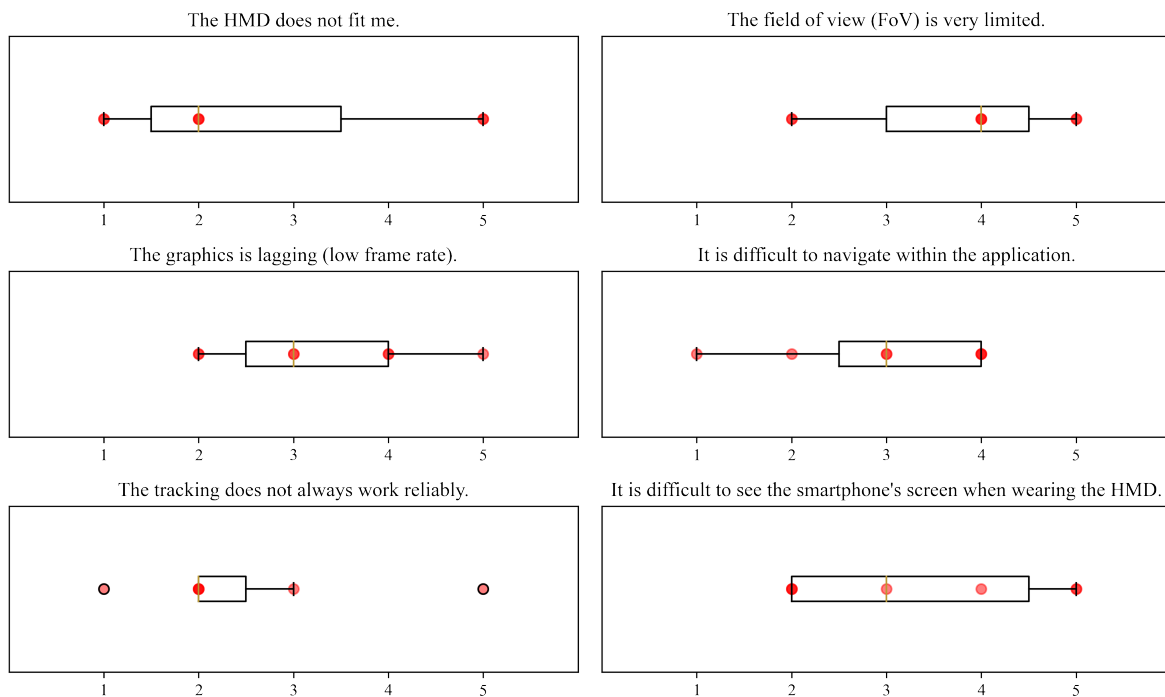


Figure 4.11.: Box plot: How significant do you think these drawbacks are in the entire system? 1 stands for "not significant at all" and 5 stands for "very significant".

It is noteworthy that users' responses to the question regarding if the AR HMD fits tended to be at two extremes: either they believed that the unsuitability of the AR HMD significantly affected the user experience, or they thought it was not a problem at all or almost not a problem. During the experiment, we also observed that participants who perceived a significant impact on the user experience due to the AR HMD had issues related to it being either too loose, causing slipping, or too tight, making it difficult to wear. This is related to individual physiological characteristics but also reflects the limitations of the Magic Leap 1 as the AR HMD.

From the box plots, it is evident that the majority of participants considered the limited

FoV of the AR HMD as a significant constraint to their user experience. According to our observations, these participants encountered difficulties in locating Robody's arms during the experiment. To keep Robody's arms in view, users often stretched them as far forward as possible, affecting the range of movement for Robody's hands. Some participants also noted that the narrow FoV made it challenging to estimate the distance between themselves and the walls. In Task 2 (turning off lights of a designated room), it was difficult for the participants to see if the lights in the room had been turned off since the FoV was mostly occupied by the light switch, hindering visibility of other elements in the scene.

A considerable number of participants agreed that lagging graphics is a significant issue, with one participant, particularly experienced with AR devices, being very sensitive to this issue. Some participants reported that hand movements were not very smooth. This could be related to our IK solver and may lead some participants to associate the experience with lagging graphics.

Several participants found navigation within the application challenging. Some participants, unfamiliar with the smartphone application and switching between different hand control methods, felt uneasy and lacked confidence during the experiment. Interestingly, two participants with the most experience using AR devices did not find this issue to be significant, suggesting that more experienced AR users can quickly adapt to our application and interaction methods.

Regarding the reliability of hand tracking, most participants did not find this to be a significant issue, with only one participant stating that the instability of hand tracking severely affected his user experience. However, this participant still chose hand tracking as his favorite control method.

In the later phase of our development, we noticed that the AR HMD display could obstruct the user's sight, making it challenging to see the content on the smartphone. Almost half of the participants considered this issue to severely affect their usage. While many participants did not find this issue very troublesome, none of the participants believed it had no impact on the user experience. Participants who considered this issue significant were observed frequently lifting their heads to see the smartphone screen through the gap beneath the AR HMD, and having difficulties in accurately clicking on the buttons on the smartphone screen. Some participants reported that this issue would become less pronounced with habitual use.

4.4.6. Task Complete Time

In the experiment, we recorded the time each participant took to complete each task. Each participant completed task 1 as the first step. Based on our observations, participants who were more confident in the system tended to use a combination of autonomous task assignment and manual steering, while those who felt pressure in navigating the application tended to conservatively rely only on manual steering. In terms of completion time, participants using a combination of task assignment and manual steering typically spent about half the time on this task compared to other participants. Participants manually steering Robody often spent a considerable amount of time navigating through narrow spaces, such as doorways, due to distance estimation issues.

4. Evaluation

In tasks 2 and 3, the order of control modes used by each participant during the experimental sessions is presented in Table 4.1.

	P1	P2	P3	P4	P5	P6	P7
Run 1	SP	HT	SM	HT	SM	SP	HT
Run 2	HT	SM	HT	SM	HT	HT	SP

Table 4.1.: Order of control modes used by participants. SP stands for "smartphone pointer", SM stands for "smartphone motion tracking" and HT stands for "hand tracking".

For tasks 2 and 3, we categorized the final investigation results by task type and created bar charts and scatter plots for analysis.

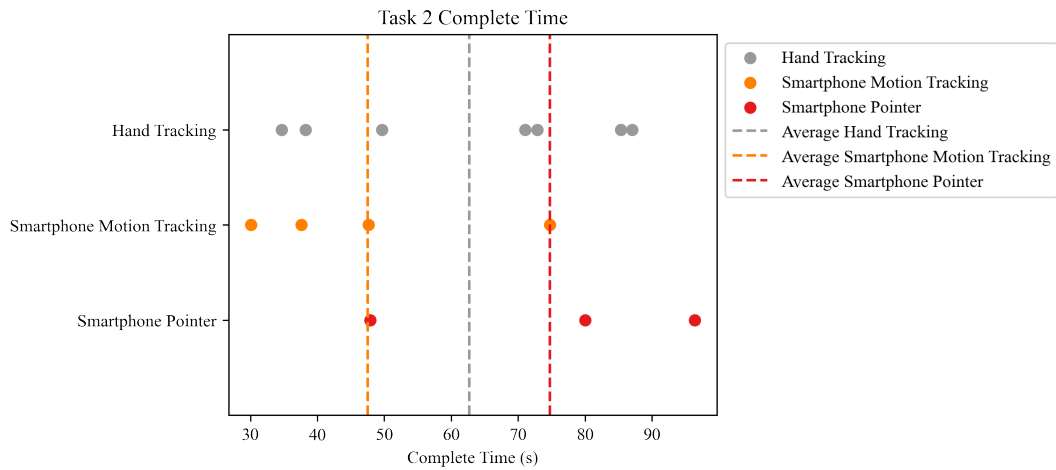


Figure 4.12.: Scatter plot: task 2 complete time.

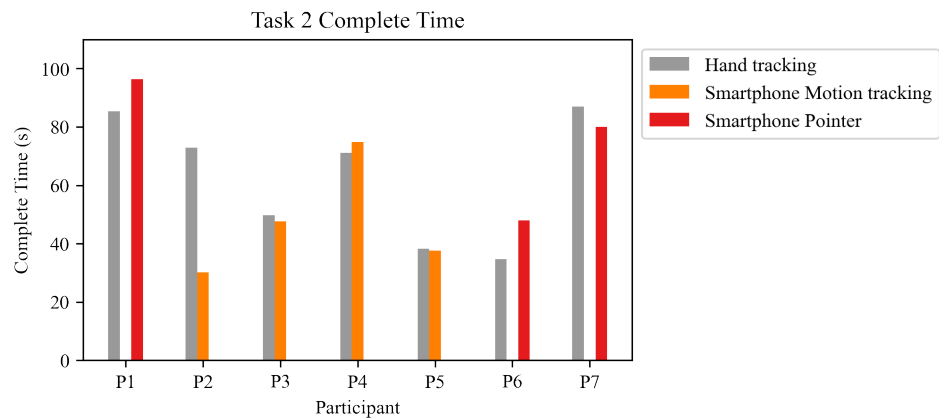


Figure 4.13.: Scatter plot: task 2 complete time.

From Figure 4.12, it can be observed that in terms of average complete time, the most

4. Evaluation

efficient (shortest time) control mode in Task 2 is smartphone motion tracking, followed by hand tracking, and lastly, smartphone pointer. From Figure 4.13, it can be observed that, except for P2, who spent significantly more time using hand tracking, participants using hand tracking and smartphone motion tracking control modes (P3, P4, P5) spent similar amounts of time in these two control modes in Task 2, indicating that the performance of these two modes is quite comparable in this context.

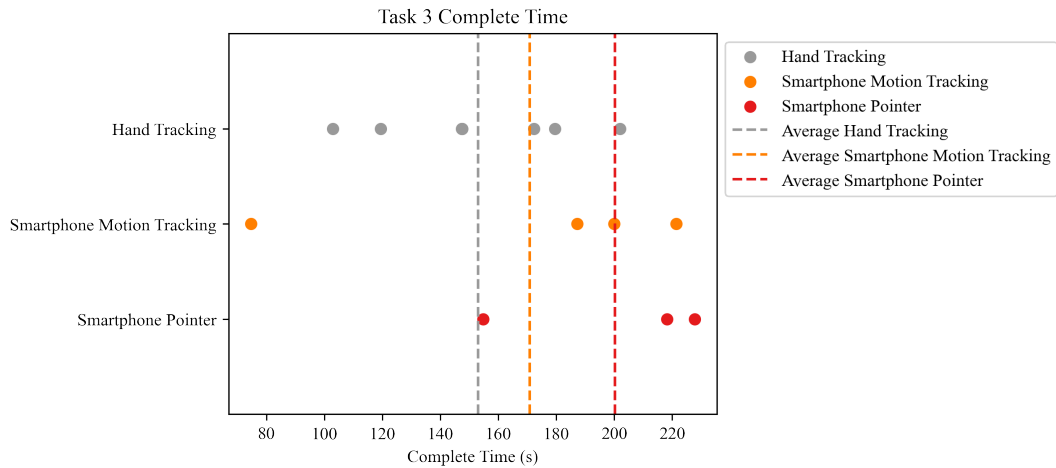


Figure 4.14.: Scatter plot: task 3 complete time.

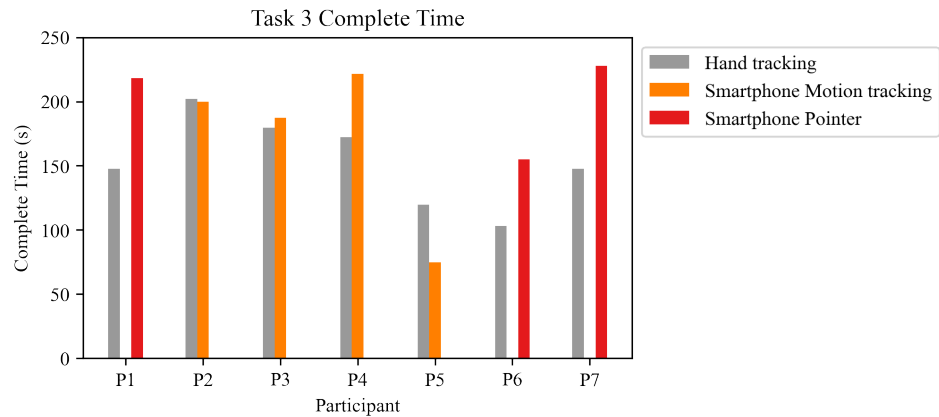


Figure 4.15.: Scatter plot: task 3 complete time.

From Figure 4.14, it can be observed that in terms of average complete time, the most efficient (shortest time) control mode in Task 3 is hand tracking, followed by smartphone motion tracking, and lastly, smartphone pointer. From Figure 4.15, it can be observed that, except for P5, who spent more time using hand tracking, all other participants spent less time or similar time using hand tracking compared to smartphone-based control modes. This result strongly indicates that in the Task 3 scenario, hand tracking is the most effective control

mode and significantly outperforms smartphone pointer control mode.

Some participants provided feedback indicating that hand tracking control mode is the most intuitive and easy-to-learn mode. Their task complete times also reflect their increased efficiency when using hand tracking for the majority of the time. At the same time, we can almost claim that the smartphone pointer control mode is the least performing mode, as it consistently took the longest time in most situations.

However, due to the inherent learning curve of the system and participants' unfamiliarity with it, the recorded task complete times may be less persuasive. A significant portion of participants needed guidance from us to understand what steps to take next, which was not our intended scenario when designing the experiment. To better validate the performance of different control modes, we may need more participants and those who are more familiar with the system in future experiments.

4.4.7. Discussion

Firstly, we can claim that our system is capable of inducing a certain level of embodiment even in the absence of high-precision 6-DoF tracking devices (AR controller). Participants generally perceived a significant sense of embodiment based on appearance. Additionally, despite most participants being unfamiliar with our system, its performance in usability ratings exceeded the average level. This demonstrates its potential when users become proficient with the system.

Some participants expressed positive views on the hand tracking control mode, finding it intuitive and easy to learn. This suggests that hand tracking technology may have potential advantages in user interface design. Furthermore, the hand tracking control mode demonstrated advantages in task complete times. In future research and development, further exploration and optimization of this control mode could be beneficial.

While the smartphone pointer control mode is robust and unaffected by lighting conditions, considering user evaluations and task complete times, there is considerable room for improvement.

Smartphone motion tracking control mode demonstrated performance in Task 2 and 3 comparable to or even better than hand tracking. With further improvements addressing its limitations, such as environmental sensitivity and the absence of haptic feedback, smartphone motion tracking could become an ideal alternative as an AR controller.

During the experiment, participants raised some issues and suggestions regarding our system.

Hardware problems were frequently mentioned. Due to our inability to successfully invoke the Magic Leap 1 API to use its built-in camera, we had to employ an additional webcam for hand tracking, leading to a series of interconnected issues.

Firstly, the design of fixing the webcam on the top of the head with an extra headphone triggered discussions. In our experiment, the webcam was attached to a headphone, and participants had to wear this headphone on top of the AR HMD to fix the webcam. Such a setup added extra weight, and several participants found it overly cumbersome. Moreover, since the webcam was fixed on the top of the head, its perspective had a significant offset

from the user's perspective. Additionally, as each user wore the webcam in a slightly different position, it's challenging to find the relative position of the webcam from the AR HMD's position, making it impractical to transform detected hand coordinate to the user-centric coordinate system. As a result, participants often needed some time to find out the webcam's detection range. Some participants suggested that fixing the webcam directly on the HMD could enhance the experience.

Secondly, there was the previously mentioned issue of limited FoV (seen in subsection 4.4.5). This caused participants difficulties in seeing Robody's arms during operation and frequently led to challenges in perceiving the surrounding environment.

Finally, some participants reported latency in the system, indicating a time lag between user actions and Robody's arm movements. This could be attributed to the development hardware setup (see 3.2b) used in the experiment process, where there might be communication delays between the PC and AR HMD. However, since we still rely on an additional webcam at present, we must use this setup to include the webcam in our system. Participants with experience using other AR devices suggested that a better AR device could alleviate the limited FoV and latency issues.

Due to the virtual nature of the scene we used, the lighting and textures in the environment fall far short of real-world effects. In the absence of detailed lighting and under the impact of limited FoV, participants experienced challenges in depth perception. Many participants found it difficult to estimate the distance between Robody's arms and target objects. Almost all participants needed to repeatedly adjust Robody's position to approach target objects.

Simultaneously, we observed that some users faced difficulties in understanding and memorizing the workflow of our system during the experiment (see Figure 3.8). Certain safety-related designs, such as the requirement for the "freeze" and "unfreeze" actions in the hand control modes, made it challenging for many participants to adapt without sufficient familiarity with our system, often needing additional reminders from us.

Participants also raised concerns about the stability of hand tracking: when participants "unfreeze" hand tracking, if the system detects an open hand, it releases the grip, causing previously grabbed items to fall. In future work, we may need to further optimize the logic and safety controls of hand tracking to fix this issue.

5. Future Work

5.1. Further Application Development

While this project has laid the foundation for controlling Robody using smartphones and AR devices, there are still many areas for improvement and further possibilities in terms of the applications.

5.1.1. Customized Control Interfaces

Different users have various preferences when using smartphones, which may be influenced by several factors:

1. **Handedness:** Users may have a preference for using their right or left hand for interactions.
2. **Hand size:** People have varying hand sizes, which may affect their comfort and flexibility when using a smartphone.
3. **Gripping style:** Individuals may hold the smartphone in different ways, such as single-handed or two-handed grip.
4. **Interaction finger:** Users use different fingers to operate on the screen, such as using thumb of the hand holding the smartphone or using the other hand's index finger.

To ensure that users with different backgrounds and preferences have the best control experience, the control interface should be customizable according to users' needs and preferences. This personalization may involve adjusting the position, size of buttons, and button mappings.

5.1.2. Multi-Platform Support

Our development and evaluation were carried out exclusively on Magic Leap 1 and an iPhone 11 running iOS. This presents significant limitations. As mentioned in subsection 3.2.1, the hardware of Magic Leap 1 is outdated and not the optimal choice for this control system. Additionally, a considerable portion of smartphone users possess Android-based devices, as opposed to iOS. Ensuring that our implementation runs on all mainstream platforms is essential to guarantee practicality and a broader scope of application.

While Unity, as a cross-platform game engine, facilitates the transition of our development to various smartphone operating systems and different AR devices, the current version of our

implementation has not been validated on other hardware devices. This could be a part of future work.

5.1.3. Known Issues

In subsection 4.4.5, some issues existing in our system have been identified and analyzed. In future work, it is imperative to fix and ameliorate these known issues based on the feedback provided by the experiment participants.

5.2. Better Hand Tracking Solution

Currently, although we have successfully implemented hand tracking based control, we acknowledge that there is room for improvement. While MediaPipe is capable of delivering reasonably good hand tracking under lightweight conditions, there is still a need to enhance accuracy and stability. In some complex scenarios, system performance might degrade, negatively affecting the user experience as users expect reliable control of Robody.

These scenarios include:

1. **Low-light conditions:** Our implementation relies on an RGB webcam. In poor lighting conditions, hand detection lacks stability and may fail to detect the user's hands or to track positional changes of the detected hands. This could potentially be improved by using depth cameras, which are equipped on some AR devices to provide additional visual information.
2. **Multiple hands in the field of view:** It is challenging to differentiate between the user's hands and those of other people when multiple hands appear in the field of view. Visual-based hand tracking may mistakenly recognize the hands of others as the user's hands.
3. **Poor performance in specific hand poses:** In some hand poses and angles, most fingers are occluded by the hand itself. In such cases, hand tracking may struggle to provide consistent and stable hand pose recognition. There are circumstances where left hands may be misidentified as right hands and vice versa.
4. **Insecurely unfrozen hand:** As mentioned in subsection 4.4.7, after reactivating hand control by "unfreezing," if an open hand posture is detected, Robody will release its grip, causing previously grabbed objects to fall.

The current setup using a webcam, which was for rapid deployment, is certainly not a long-term solution. Future development should explore the use of cameras equipped on AR headset or cameras with a larger field of view to ameliorate the hand tracking capabilities.

In future work, we can explore more advanced hand tracking solutions, including better hardware and better algorithms, to achieve more stable and reliable results.

5.3. Better Smartphone Control Method

We have proposed two smartphone-based control methods that effectively allow users to control Robody's arms. However, these control methods still have limitations. For example, although the smartphone motion tracking can track the movement of the user's hand reasonably accurately, it still relies on camera input and has degraded performance in low-light environments and surroundings lacking features suitable for SLAM tracking. This may result in occasional positional drifts.

One possible solution could involve displaying a marker on the smartphone screen that can be tracked by the AR headset's camera, allowing for the calculation of the smartphone's relative position to the AR headset as a means of correcting the smartphone's position.

Additionally, the process of calibrating the smartphone's position and orientation before using it for control could be made a more automated and user-friendly process.

Furthermore, when users wear AR HMD, their field of view can be occluded by the displayed content within the AR HMD, making it challenging for users to observe the content on the smartphone screen. Buttons on the smartphone screen cannot effectively provide haptic feedback, requiring users to spend additional effort to identify the smartphone screen's content. Using physical buttons on the smartphone as an alternative may be one solution (such as volume keys).

5.4. Voice control

When performing hand tracking, it's inconvenient for users to perform additional actions with their hands. Extra movements may undesirably be captured by the camera, leading to unexpected actions of Robody. However, not triggering additional command to stop tracking will keep hand tracking active. This creates a control flow conflict. Our current solution is to use gaze with dwell-time for additional operations in the hand tracking control mode, such as freezing hand movements and returning to the main menu. This way, hand movement is avoided, but head movement is still required to select buttons. Previous work has shown that gaze with dwell-time is not always the best choice for executing commands [45].

Work of Jorgenson et al. used voice control to freeze and thaw robot movements [81]. This could be a direction for our future work: adding voice control to our control system to better avoid control flow conflicts through multimodal control.

5.5. Integration with Physical Robody

In subsection 4.4.1, we discussed some of the remaining issues with our integration. In addition to solving these problems, we can also identify some future directions for our work. Currently, our integration is limited to the arms. Future work should include integrating head posture and finger flexing into the physical Robody. Transmitting the perspective captured by Robody's camera to the AR HMD should also be handled.

6. Conclusion

In this thesis, we first conducted a literature review on various relevant aspects and revisited prior efforts to achieve remote embodiment of Robody. We extended the teleoperation system for Robody from previous works, aiming to create a comprehensive Robody control platform using a smartphone and an AR HMD.

We developed a smartphone application possessing various functionalities related to Robody, such as remote monitoring, autonomous task assignment, and serving as a controller. The smartphone application is designed to allow the smartphone to replace specialized AR controllers for interaction with the AR HMD and control of Robody's hands. We proposed two smartphone-based hand control methods: smartphone pointer and smartphone motion tracking. Additionally, we utilized MediaPipe's hand landmark detection function to develop a vision-based hand tracking control method for intuitive control of Robody's hands through a monocular RGB camera.

We reconstructed a virtual clinical environment to enable virtual Robody to interact with scenes and objects.


As part of the evaluation, after successfully realizing control of Robody in simulation software, we integrated our control methods onto the physical Robody and successfully controlled its arm.

In the other part of the evaluation, we conducted a user study to assess the system usability, embodiment, and performance of each control method. Based on the collected feedback, our system's usability exceeded the average level, and it also demonstrated the ability to induce a certain level of embodiment. The hand tracking control mode performed the best in the experiment, while the smartphone pointer control mode performed the worst. The limitations of the Magic Leap 1 hardware were emphasized by many participants. In the evaluation, the hand tracking-based control showed its future potential, while smartphone-based control methods still have room for improvement.

We concluded by summarizing future directions, including the discussion of known issues and potential new directions in the long-run.

A. Appendix

A.1. Consent Form



Forschungsgruppe Augmented Reality (FAR)
Prof. Dr. Gudrun Klinker

Technische Universität München

Consent Form

DESCRIPTION: You are invited to participate in the user study of a master thesis “Integration of Mixed Reality and Touchscreen Interfaces for Humanoid Robot Embodiment in a Virtual Clinical Setting” . This user study is about assessing the usability and the induced sense of embodiment of our implementation.

TIME INVOLVEMENT: The user study involves experiencing our implemented control method and conducting several small tasks in the virtual environment using a smartphone and an augmented reality head-mounted-display (AR HMD), following by filling out 3 questionnaires (30 questions in total). It will take about 20-25 minutes.

DATA COLLECTION: Demographics, in-task used time will be collected. For this experiment, all data is collected pseudo-anonymously. The collected data will be stored in the server of Google Form and will be deleted in 3 months.

RISKS The study involves wearing an AR HMD, which may induce motion sickness in certain individuals. Apart from that, there are no known risks associated with that study. None of the data which we collect can be traced to a specific individual.

PAYMENT: Each participant will receive a compensation of 10 euros and our sincere gratitude. Additionally, you will have the opportunity to take part in an interesting project.

PARTICIPANT'S RIGHTS: If you have read this form and have decided to participate in this project, please understand that although your **participation is voluntary**, we would like your involvement as much as you can afford. That being said, you have the **right to withdraw your consent or discontinue participation at any time without penalty or loss of benefits to which you are otherwise entitled**. The results of this research study may be presented at academic meetings and included in a master thesis. Your identity is not disclosed unless we directly inform and ask for your permission.

CONTACT INFORMATION: If you have any questions, concerns or complaints about this research, its procedures, risks and benefits, contact the following persons:
Yinfeng Yu, Master Student (yinfeng.yu@tum.de)
Christian Eichhorn (christian.eichhorn@tum.de)

- I have read and understood the consent form. I understand the purpose of this user study and agree to participate.
- I understand that I may terminate my participation in the study at any time.
- I accept the recording of my user data and am also aware that excerpts from interviews will be kept anonymous and maybe included in a publication.

By signing this document, I confirm that I agree to the terms and conditions.

Name

Signature, Date

A.2. Questionnaire 1 (System Usability Scale)

This questionnaire is the system usability scale, designed to assess the usability of our implementation. Please tell us about your experience using our control system. We appreciate your time and feedback!

1. I think that I would like to use this system frequently.

Strongly disagree Disagree Neutral Agree Strongly Agree

2. I found the system unnecessarily complex.

Strongly disagree Disagree Neutral Agree Strongly Agree

3. I thought the system was easy to use.

Strongly disagree Disagree Neutral Agree Strongly Agree

4. I think that I would need the support of a technical person to be able to use this system.

Strongly disagree Disagree Neutral Agree Strongly Agree

5. I found the various functions in this system were well integrated.

Strongly disagree Disagree Neutral Agree Strongly Agree

6. I thought there was too much inconsistency in this system.

Strongly disagree Disagree Neutral Agree Strongly Agree

7. I would imagine that most people would learn to use this system very quickly.

Strongly disagree Disagree Neutral Agree Strongly Agree

8. I found the system very cumbersome to use.

Strongly disagree Disagree Neutral Agree Strongly Agree

9. I felt very confident using the system.

Strongly disagree Disagree Neutral Agree Strongly Agree

10. I needed to learn a lot of things before I could get going with this system.

Strongly disagree Disagree Neutral Agree Strongly Agree

A.3. Questionnaire 2 (Embodiment Questionnaire)

This questionnaire is a standardized embodiment questionnaire originally proposed by Peck and Gonzalez-Franco. It is designed to measure the level of induced embodiment. Please note that it is a lengthy questionnaire, and we greatly appreciate your patience!

1. I felt out of my body.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

2. I felt as if my (real) arm were drifting toward the virtual arm or as if the virtual arm were drifting toward my (real) arm.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

3. I felt as if the movements of the virtual arm were influencing my own movement.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

4. It felt as if my (real) arm were turning into an "avatar" arm.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

5. At some point it felt as if my real arm was starting to take on the posture or shape of the virtual arm that I saw.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

6. I felt like I was wearing different clothes from when I came to the experiment place.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

7. I felt as if my arm had changed.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

8. I felt colliding with a wall when I saw the virtual arm collide with a wall.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

9. I felt that my own arm could be affected by the surrounding environment.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

10. I felt as if the virtual arm was my arm.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

11. At some point it felt that the virtual arm resembled my own (real) arm, in terms of shape, skin tone or other visual features.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

12. I felt as if my arm was located where I saw the virtual arm.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

13. I felt like I could control the virtual arm as if it was my own arm.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

14. It seemed as if I felt the touch of the grabbed object in the location where I saw the virtual hand touched.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

15. It seemed as if the touch I felt was caused by the grabbed object touching the virtual hand.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

16. It seemed as if my hand was touching the grabbable object.

Strongly disagree 1 2 3 4 5 6 7 Strongly Agree
Neutral

A.4. Questionnaire 3

This questionnaire includes several questions specific to our implementation. While only the first two questions are mandatory, we greatly appreciate your full response!

1. How much prior experience do you have with AR or VR?

Never 1 2 3 4 5 I am an expert

2. Which control methods do you prefer the most and the least?

	Smartphone pointer	Smartphone motion tracking	Hand tracking
Favourite control method	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Least favourite control method	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3. (Optional) In the entire system, what do you consider to be the most serious drawbacks?

	Very significant	Significant	Moderate	Not very significant	Not significant at all
The HMD does not fit me.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The field of view (FoV) is very limited.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The graphics is lagging (low frame rate).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
It is difficult to navigate within the application.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The tracking does not always work reliably.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

A. Appendix

It is difficult to see the smart-phone's screen when wearing the HMD.

4. (Optional) Do you have any suggestions on any of the control modes or the entire system?

List of Figures

2.1. Existing robots in healthcare	9
2.2. Model of a digital twin	14
2.3. Different HRI styles	22
2.4. VR interface developed by Jorgensen et al.	26
2.5. Mobile phone/Smartphone interfaces for controlling robots	30
2.6. Vision-based hand tracking for robot hand control	32
3.1. MediaPipe hand tracking solution architecture	45
3.2. Device communication system	47
3.3. Control page interface	48
3.4. Magic Leap 1 controller	49
3.5. Illustration of main menu button selection	50
3.6. Menu UIs	51
3.7. Illustration of calibration operation	52
3.8. Workflow of the control system	53
3.9. Monitoring page interface	54
3.10. Task assignment page interface	55
3.11. Hand tracking control mode concept	56
3.12. MediaPipe hand landmark model	56
3.13. Demonstration of palm orientation calculation	57
3.14. Illustration of the hand tracking control mode	58
3.15. Hand tracking control mode concept	59
3.16. Illustration of the smartphone pointer control mode	59
3.17. Hand control interface (Smartphone Pointer Mode)	60
3.18. Hand tracking control mode concept	61
3.19. Illustration of the smartphone motion tracking control mode	62
3.20. Hand model introduced by Lee et al.	64
3.21. Demonstration of finger joint angle calculation	65
4.1. Software setup of the ROS simulation.	69
4.2. Demonstration of ROS simulation integration	70
4.3. Successful integration with physical Robody	70
4.4. Experiment flow	73
4.5. Pie chart: prior experience with AR/VR	76
4.6. Box plot: system usability score	77
4.7. Box plot: sub-scales of embodiment	78

List of Figures

4.8. Box plot: embodiment score	78
4.9. Scatter plot: correlation between system usability scores and embodiment scores	79
4.10. Bar chart: Which control methods do you like the most and the least?	79
4.11. Box plot: How significant do you think these drawbacks are in the entire system?	80
4.12. Scatter plot: task 2 complete time	82
4.13. Bar chart: task 2 complete time	82
4.14. Scatter plot: task 3 complete time	83
4.15. Bar chart: task 3 complete time	83

List of Tables

- 3.1. Comparison of different hand control strategies 42
- 4.1. Order of control modes used by participants 82

Glossary

anosmia is the inability to smell. 1

autism spectrum disorder (ASD) is a developmental disability caused by differences in the brain. People with ASD often have problems with social communication and interaction, and restricted or repetitive behaviors or interests ¹. 6, 9

dysgeusia is a distortion of the sense of taste. 1

¹Signs and Symptoms of Autism Spectrum Disorder. <https://www.cdc.gov/ncbddd/autism/signs.html>.

Bibliography

- [1] U. Karlsson and C.-J. Fraenkel. “Covid-19: risks to healthcare workers and their families”. In: *BMJ* (Oct. 2020), p. m3944. ISSN: 1756-1833. DOI: 10.1136/bmj.m3944. URL: <https://www.bmj.com/lookup/doi/10.1136/bmj.m3944>.
- [2] N. Magnavita, G. Tripepi, and R. R. Di Prinzio. “Symptoms in Health Care Workers during the COVID-19 Epidemic. A Cross-Sectional Survey”. In: *International Journal of Environmental Research and Public Health* 17.14 (July 2020), p. 5218. ISSN: 1660-4601. DOI: 10.3390/ijerph17145218. URL: <https://www.mdpi.com/1660-4601/17/14/5218>.
- [3] S. Bandyopadhyay, R. E. Baticulon, M. Kadhun, M. Alser, D. K. Ojuka, Y. Badereddin, A. Kamath, S. A. Parepalli, G. Brown, S. Iharchane, S. Gandino, Z. Markovic-Obiago, S. Scott, E. Manirambona, A. Machhada, A. Aggarwal, L. Benazaize, M. Ibrahim, D. Kim, I. Tol, E. H. Taylor, A. Knighton, D. Bbaale, D. Jasim, H. Alghoul, H. Reddy, H. Abuelgasim, K. Saini, A. Sigler, L. Abuelgasim, M. Moran-Romero, M. Kumarendran, N. A. Jamie, O. Ali, R. Sudarshan, R. Dean, R. Kissyova, S. Kelzang, S. Roche, T. Ahsan, Y. Mohamed, A. M. Dube, G. P. Gwini, R. Gwokyaala, R. Brown, M. R. K. K. Papon, Z. Li, S. S. Ruzats, S. Charuvila, N. Peter, K. Khalidy, N. Moyo, O. Alser, A. Solano, E. Robles-Perez, A. Tariq, M. Gaddah, S. Kolovos, F. C. Muchemwa, A. Saleh, A. Gosman, R. Pinedo-Villanueva, A. Jani, and R. Khundkar. “Infection and mortality of healthcare workers worldwide from COVID-19: a systematic review”. In: *BMJ Global Health* 5.12 (Dec. 2020), e003097. ISSN: 2059-7908. DOI: 10.1136/bmjgh-2020-003097. URL: <https://gh.bmj.com/lookup/doi/10.1136/bmjgh-2020-003097>.
- [4] M. Kyrarini, F. Lygerakis, A. Rajavenkatanarayanan, C. Sevastopoulos, H. R. Nambiappan, K. K. Chaitanya, A. R. Babu, J. Mathew, and F. Makedon. “A Survey of Robots in Healthcare”. In: *Technologies* 9.1 (Jan. 2021), p. 8. ISSN: 2227-7080. DOI: 10.3390/technologies9010008. URL: <https://www.mdpi.com/2227-7080/9/1/8>.
- [5] K. J. Vänni and S. E. Salin. “A Need for Service Robots Among Health Care Professionals in Hospitals and Housing Services”. In: *Social Robotics*. Ed. by A. Kheddar, E. Yoshida, S. S. Ge, K. Suzuki, J.-J. Cabibihan, F. Eyssel, and H. He. Vol. 10652. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 178–187. ISBN: 978-3-319-70021-2 978-3-319-70022-9. DOI: 10.1007/978-3-319-70022-9_18. URL: http://link.springer.com/10.1007/978-3-319-70022-9_18.
- [6] L. D. Riek. “Healthcare robotics”. In: *Communications of the ACM* 60.11 (Oct. 2017), pp. 68–78. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3127874. URL: <https://dl.acm.org/doi/10.1145/3127874>.

- [7] A. van Wynsberghe. "Designing Robots for Care: Care Centered Value-Sensitive Design". In: *Science and Engineering Ethics* 19.2 (June 2013), pp. 407–433. ISSN: 1353-3452, 1471-5546. DOI: 10.1007/s11948-011-9343-6. URL: <http://link.springer.com/10.1007/s11948-011-9343-6>.
- [8] D. Feil-Seifer and M. J. Matarić. "Defining Socially Assistive Robotics". In: *Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics*. 2005.
- [9] G. Canal, G. Alenya, and C. Torras. "A taxonomy of preferences for physically assistive robots". In: *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. Lisbon: IEEE, Aug. 2017, pp. 292–297. ISBN: 978-1-5386-3518-6. DOI: 10.1109/ROMAN.2017.8172316. URL: <http://ieeexplore.ieee.org/document/8172316/>.
- [10] G. Canal, G. Alenyà, and C. Torras. "Personalization Framework for Adaptive Robotic Feeding Assistance". In: *Social Robotics*. Ed. by A. Agah, J.-J. Cabibihan, A. M. Howard, M. A. Salichs, and H. He. Vol. 9979. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 22–31. ISBN: 978-3-319-47436-6 978-3-319-47437-3. DOI: 10.1007/978-3-319-47437-3_3. URL: http://link.springer.com/10.1007/978-3-319-47437-3_3.
- [11] G. Chance, A. Camilleri, B. Winstone, P. Caleb-Solly, and S. Dogramadzi. "An assistive robot to support dressing - strategies for planning and error handling". In: *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. Singapore, Singapore: IEEE, June 2016, pp. 774–780. ISBN: 978-1-5090-3287-7. DOI: 10.1109/BIOROB.2016.7523721. URL: <http://ieeexplore.ieee.org/document/7523721/>.
- [12] R. Y. Kim. "Robots in Healthcare". In: *Interactions* 30.1 (Jan. 2023), pp. 48–51. ISSN: 1072-5520. DOI: 10.1145/3575871. URL: <https://doi.org/10.1145/3575871>.
- [13] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi. "Introduction". In: *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg, 2014, pp. 1–17. DOI: 10.1007/978-3-642-54536-8_1. URL: https://doi.org/10.1007/978-3-642-54536-8_1.
- [14] A. Khan and Y. Anwar. "Robots in Healthcare: A Survey". In: *Advances in Computer Vision*. Ed. by K. Arai and S. Kapoor. Vol. 944. Series Title: Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2020, pp. 280–292. ISBN: 978-3-030-17797-3 978-3-030-17798-0. DOI: 10.1007/978-3-030-17798-0_24. URL: http://link.springer.com/10.1007/978-3-030-17798-0_24.
- [15] J. Fink. "Anthropomorphism and Human Likeness in the Design of Robots and Human-Robot Interaction". In: *Social Robotics*. Ed. by D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, S. S. Ge, O. Khatib, J.-J. Cabibihan, R. Simmons, and M.-A. Williams. Vol. 7621. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 199–208. ISBN: 978-3-642-34102-1 978-3-642-34103-8. DOI: 10.1007/978-3-642-34103-8_20. URL: https://link.springer.com/10.1007/978-3-642-34103-8_20.

- [16] B. R. Duffy. "Anthropomorphism and the social robot". In: *Robotics and Autonomous Systems* 42.3-4 (Mar. 2003), pp. 177–190. ISSN: 09218890. DOI: 10.1016/S0921-8890(02)00374-3. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0921889002003743>.
- [17] M. Sato, Y. Yasuhara, K. Osaka, H. Ito, M. J. S. Dino, I. L. Ong, Y. Zhao, and T. Tanioka. "Rehabilitation care with Pepper humanoid robot: A qualitative case study of older patients with schizophrenia and/or dementia in Japan". In: *Enfermería Clínica* 30 (2020). Our Lady of Fatima University's Research Development and Innovation Conference 2019, pp. 32–36. ISSN: 1130-8621. DOI: <https://doi.org/10.1016/j.enfcli.2019.09.021>. URL: <https://www.sciencedirect.com/science/article/pii/S1130862119305807>.
- [18] A. K. Pandey and R. Gelin. "A Mass-Produced Sociable Humanoid Robot: Pepper: The First Machine of Its Kind". In: *IEEE Robotics & Automation Magazine* 25.3 (Sept. 2018), pp. 40–48. ISSN: 1070-9932, 1558-223X. DOI: 10.1109/MRA.2018.2833157. URL: <https://ieeexplore.ieee.org/document/8409927/>.
- [19] F. Carros, J. Meurer, D. Löffler, D. Unbehaun, S. Matthies, I. Koch, R. Wieching, D. Randall, M. Hassenzahl, and V. Wulf. "Exploring Human-Robot Interaction with the Elderly: Results from a Ten-Week Case Study in a Care Home". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, Apr. 2020, pp. 1–12. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376402. URL: <https://dl.acm.org/doi/10.1145/3313831.3376402>.
- [20] U. Qidwai, S. B. A. Kashem, and O. Conor. "Humanoid Robot as a Teacher's Assistant: Helping Children with Autism to Learn Social and Academic Skills". In: *Journal of Intelligent & Robotic Systems* 98.3-4 (June 2020), pp. 759–770. ISSN: 0921-0296, 1573-0409. DOI: 10.1007/s10846-019-01075-1. URL: <http://link.springer.com/10.1007/s10846-019-01075-1>.
- [21] Diligent Robotics. *Moxi*. URL: <https://www.diligentrobots.com/moxi>.
- [22] A. Sharkey and N. Sharkey. "Granny and the robots: ethical issues in robot care for the elderly". In: *Ethics and Information Technology* 14.1 (Mar. 2012), pp. 27–40. ISSN: 1388-1957, 1572-8439. DOI: 10.1007/s10676-010-9234-6. URL: <http://link.springer.com/10.1007/s10676-010-9234-6>.
- [23] R. Sparrow and L. Sparrow. "In the hands of machines? The future of aged care". In: *Minds and Machines* 16.2 (Oct. 2006), pp. 141–161. ISSN: 0924-6495, 1572-8641. DOI: 10.1007/s11023-006-9030-6. URL: <http://link.springer.com/10.1007/s11023-006-9030-6>.
- [24] H. Choi, C. Crump, C. Duriez, A. Elmquist, G. Hager, D. Han, F. Hearl, J. Hodgins, A. Jain, F. Leve, C. Li, F. Meier, D. Negrut, L. Righetti, A. Rodriguez, J. Tan, and J. Trinkle. "On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward". In: *Proceedings of the National Academy of Sciences* 118.1 (Jan.

- 2021), e1907856118. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1907856118. URL: <https://pnas.org/doi/full/10.1073/pnas.1907856118>.
- [25] M. Duguleana, F. G. Barbuceanu, and G. Mogan. "Evaluating Human-Robot Interaction during a Manipulation Experiment Conducted in Immersive Virtual Reality". In: *Virtual and Mixed Reality - New Trends*. Ed. by R. Shumaker. Vol. 6773. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 164–173. ISBN: 978-3-642-22020-3 978-3-642-22021-0. DOI: 10.1007/978-3-642-22021-0_19. URL: http://link.springer.com/10.1007/978-3-642-22021-0_19.
- [26] R. Serban, M. Taylor, D. Negrut, and A. Tasora. "Chrono::Vehicle: template-based ground vehicle modelling and simulation". In: *International Journal of Vehicle Performance* 5 (2019). DOI: 10.1504/IJVP.2019.097096.
- [27] E. Franti, D. Tufis, S. Goschin, M. Dascalu, P. Milea, G. Stefan, T. Balan, C. Slav, and R. Demco. "Virtual environment for robots interfaces design and testing". In: *CAS 2005 Proceedings. 2005 International Semiconductor Conference, 2005*. Vol. 2. Sinaia, Romania: IEEE, 2005, pp. 463–466. ISBN: 978-0-7803-9214-4. DOI: 10.1109/SMICND.2005.1558827. URL: <http://ieeexplore.ieee.org/document/1558827/>.
- [28] S. Kiesler, A. Powers, S. R. Fussell, and C. Torrey. "Anthropomorphic Interactions with a Robot and Robot-like Agent". In: *Social Cognition* 26.2 (Apr. 2008), pp. 169–181. ISSN: 0278-016X. DOI: 10.1521/soco.2008.26.2.169. URL: <http://guilfordjournals.com/doi/10.1521/soco.2008.26.2.169>.
- [29] M. Singh, E. Fuenmayor, E. Hinchy, Y. Qiao, N. Murray, and D. Devine. "Digital Twin: Origin to Future". In: *Applied System Innovation* 4.2 (May 2021), p. 36. ISSN: 2571-5577. DOI: 10.3390/asi4020036. URL: <https://www.mdpi.com/2571-5577/4/2/36>.
- [30] B. R. Barricelli, E. Casiraghi, and D. Fogli. "A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications". In: *IEEE Access* 7 (2019), pp. 167653–167671. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2953499. URL: <https://ieeexplore.ieee.org/document/8901113/>.
- [31] M. Grieves. *Digital twin: manufacturing excellence through virtual factory replication*. 2014.
- [32] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks. "Characterising the Digital Twin: A systematic literature review". In: *CIRP Journal of Manufacturing Science and Technology* 29 (May 2020), pp. 36–52. ISSN: 17555817. DOI: 10.1016/j.cirpj.2020.02.002. URL: <https://linkinghub.elsevier.com/retrieve/pii/S1755581720300110>.
- [33] Q. Qi and F. Tao. "Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison". In: *IEEE Access* 6 (2018), pp. 3585–3593. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2793265. URL: <http://ieeexplore.ieee.org/document/8258937/>.
- [34] F. Tao. "Digital twin-driven product design, manufacturing and service with big data". In: *Int J Adv Manuf Technol* (2018).

- [35] E. Glaessgen and D. Stargel. "The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles". In: *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* & *20th AIAA/ASME/AHS Adaptive Structures Conference* & *14th AIAA*. Honolulu, Hawaii: American Institute of Aeronautics and Astronautics, Apr. 2012. ISBN: 978-1-60086-937-2. DOI: 10.2514/6.2012-1818. URL: <http://arc.aiaa.org/doi/abs/10.2514/6.2012-1818>.
- [36] W. Shen, C. Yang, and L. Gao. "Address business crisis caused by COVID-19 with collaborative intelligent manufacturing technologies". In: *IET Collaborative Intelligent Manufacturing 2.2* (June 2020), pp. 96–99. ISSN: 2516-8398, 2516-8398. DOI: 10.1049/iet-cim.2020.0041. URL: <https://onlinelibrary.wiley.com/doi/10.1049/iet-cim.2020.0041>.
- [37] T. Kaarlela, S. Pieska, and T. Pitkaaho. "Digital Twin and Virtual Reality for Safety Training". In: *2020 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*. Mariehamn, Finland: IEEE, Sept. 2020, pp. 000115–000120. ISBN: 978-1-72818-213-1. DOI: 10.1109/CogInfoCom50765.2020.9237812. URL: <https://ieeexplore.ieee.org/document/9237812/>.
- [38] M. Grieves and J. Vickers. "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems". In: *Transdisciplinary Perspectives on Complex Systems*. Ed. by F.-J. Kahlen, S. Flumerfelt, and A. Alves. Cham: Springer International Publishing, 2017, pp. 85–113. ISBN: 978-3-319-38754-3 978-3-319-38756-7. DOI: 10.1007/978-3-319-38756-7_4. URL: http://link.springer.com/10.1007/978-3-319-38756-7_4.
- [39] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn. "Digital Twin in manufacturing: A categorical literature review and classification". In: *IFAC-PapersOnLine* 51.11 (2018), pp. 1016–1022. ISSN: 24058963. DOI: 10.1016/j.ifacol.2018.08.474. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2405896318316021>.
- [40] F. Tao, Q. Qi, L. Wang, and A. Nee. "Digital Twins and Cyber-Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison". In: *Engineering* 5.4 (Aug. 2019), pp. 653–661. ISSN: 20958099. DOI: 10.1016/j.eng.2019.01.014. URL: <https://linkinghub.elsevier.com/retrieve/pii/S209580991830612X>.
- [41] E. Ferko, A. Bucaioni, P. Pelliccione, and M. Behnam. "Standardisation in Digital Twin Architectures in Manufacturing". In: *2023 IEEE 20th International Conference on Software Architecture (ICSA)*. L'Aquila, Italy: IEEE, Mar. 2023, pp. 70–81. ISBN: 9798350397499. DOI: 10.1109/ICSA56044.2023.00015. URL: <https://ieeexplore.ieee.org/document/10092613/>.
- [42] D. R. Olsen and T. Nielsen. "Laser pointer interaction". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Seattle Washington USA: ACM, Mar. 2001, pp. 17–22. ISBN: 978-1-58113-327-1. DOI: 10.1145/365024.365030. URL: <https://dl.acm.org/doi/10.1145/365024.365030>.

- [43] J. Seifert, A. Bayer, and E. Rukzio. "PointerPhone: Using Mobile Phones for Direct Pointing Interactions with Remote Displays". In: *Human-Computer Interaction – INTERACT 2013*. Ed. by D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, P. Kotzé, G. Marsden, G. Lindgaard, J. Wesson, and M. Winckler. Vol. 8119. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 18–35. ISBN: 978-3-642-40476-4 978-3-642-40477-1. DOI: 10.1007/978-3-642-40477-1_2. URL: http://link.springer.com/10.1007/978-3-642-40477-1_2.
- [44] A. Bragdon, R. DeLine, K. Hinckley, and M. R. Morris. "Code Space: Touch + Air Gesture Hybrid Interactions for Supporting Developer Meetings". In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ITS '11. Kobe, Japan: Association for Computing Machinery, 2011, pp. 212–221. ISBN: 9781450308717. DOI: 10.1145/2076354.2076393. URL: <https://doi.org/10.1145/2076354.2076393>.
- [45] R. J. K. Jacob. "What you look at is what you get: eye movement-based interaction techniques". In: *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90*. Seattle, Washington, United States: ACM Press, 1990, pp. 11–18. ISBN: 978-0-201-50932-8. DOI: 10.1145/97243.97246. URL: <http://portal.acm.org/citation.cfm?doid=97243.97246>.
- [46] L. E. Sibert and R. J. K. Jacob. "Evaluation of eye gaze interaction". In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. The Hague The Netherlands: ACM, Apr. 2000, pp. 281–288. ISBN: 978-1-58113-216-8. DOI: 10.1145/332040.332445. URL: <https://dl.acm.org/doi/10.1145/332040.332445>.
- [47] S. Zhai, C. Morimoto, and S. Ihde. "Manual and gaze input cascaded (MAGIC) pointing". In: *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99*. Pittsburgh, Pennsylvania, United States: ACM Press, 1999, pp. 246–253. ISBN: 978-0-201-48559-2. DOI: 10.1145/302979.303053. URL: <http://portal.acm.org/citation.cfm?doid=302979.303053>.
- [48] H. Graf and K. Jung. "The smartphone as a 3D input device". In: *2012 IEEE Second International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*. Berlin, Germany: IEEE, Sept. 2012, pp. 254–257. ISBN: 978-1-4673-1547-0 978-1-4673-1546-3 978-1-4673-1545-6. DOI: 10.1109/ICCE-Berlin.2012.6336487. URL: <http://ieeexplore.ieee.org/document/6336487/>.
- [49] T. Vajk, P. Coulton, W. Bamford, and R. Edwards. "Using a Mobile Phone as a "Wii-like" Controller for Playing Games on a Large Public Display". In: *International Journal of Computer Games Technology 2008 (2008)*, pp. 1–6. ISSN: 1687-7047, 1687-7055. DOI: 10.1155/2008/539078. URL: <http://www.hindawi.com/journals/ijcgt/2008/539078/>.
- [50] N. Katakakis and M. Hori. "Mobile devices as multi-DOF controllers". In: *2010 IEEE Symposium on 3D User Interfaces (3DUI)*. Waltham, MA, USA: IEEE, Mar. 2010, pp. 139–140. ISBN: 978-1-4244-6846-1. DOI: 10.1109/3DUI.2010.5444700. URL: <http://ieeexplore.ieee.org/document/5444700/>.

- [51] S. Stellmach and R. Dachsel. "Look & touch: gaze-supported target acquisition". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Austin Texas USA: ACM, May 2012, pp. 2981–2990. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2208709. URL: <https://dl.acm.org/doi/10.1145/2207676.2208709>.
- [52] K. Pfeuffer, J. Alexander, and H. Gellersen. "Gaze+touch vs. Touch: What's the Trade-off When Using Gaze to Extend Touch to Remote Displays?" In: *Human-Computer Interaction – INTERACT 2015*. Ed. by J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, and M. Winckler. Vol. 9297. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 349–367. ISBN: 978-3-319-22667-5 978-3-319-22668-2. DOI: 10.1007/978-3-319-22668-2_27. URL: http://link.springer.com/10.1007/978-3-319-22668-2_27.
- [53] T. Babic, H. Reiterer, and M. Haller. "Pocket6: A 6DoF Controller Based On A Simple Smartphone Application". In: *Proceedings of the Symposium on Spatial User Interaction*. Berlin Germany: ACM, Oct. 2018, pp. 2–10. ISBN: 978-1-4503-5708-1. DOI: 10.1145/3267782.3267785. URL: <https://dl.acm.org/doi/10.1145/3267782.3267785>.
- [54] J. Rekimoto. "Matrix: a realtime object identification and registration method for augmented reality". In: *Proceedings. 3rd Asia Pacific Computer Human Interaction (Cat. No.98EX110)*. Shonan Village Center, Japan: IEEE Comput. Soc, 1998, pp. 63–68. ISBN: 978-0-8186-8347-3. DOI: 10.1109/APCHI.1998.704151. URL: <http://ieeexplore.ieee.org/document/704151/>.
- [55] R. F. Salas-Moreno, B. Glocken, P. H. J. Kelly, and A. J. Davison. "Dense planar SLAM". In: *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. Munich, Germany: IEEE, Sept. 2014, pp. 157–164. ISBN: 978-1-4799-6184-9. DOI: 10.1109/ISMAR.2014.6948422. URL: <http://ieeexplore.ieee.org/document/6948422/>.
- [56] Yun Chan Cho and Jae Wook Jeon. "Remote robot control system based on DTMF of mobile phone". In: *2008 6th IEEE International Conference on Industrial Informatics*. Daejeon, South Korea: IEEE, July 2008, pp. 1441–1446. ISBN: 978-1-4244-2170-1. DOI: 10.1109/INDIN.2008.4618331. URL: <http://ieeexplore.ieee.org/document/4618331/>.
- [57] J. Rekimoto and K. Nagao. "The world through the computer: computer augmented interaction with real world environments". In: *Proceedings of the 8th annual ACM symposium on User interface and software technology*. Pittsburgh Pennsylvania USA: ACM, Dec. 1995, pp. 29–36. ISBN: 978-0-89791-709-4. DOI: 10.1145/215585.215639. URL: <https://dl.acm.org/doi/10.1145/215585.215639>.
- [58] O. Phaijit, M. Obaid, C. Sammut, and W. Johal. "A Taxonomy of Functional Augmented Reality for Human-Robot Interaction". In: *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Sapporo, Japan: IEEE, Mar. 2022, pp. 294–303. ISBN: 978-1-66540-731-1. DOI: 10.1109/HRI53351.2022.9889622. URL: <https://ieeexplore.ieee.org/document/9889622/>.

- [59] K. Reed, M. Peshkin, M. Hartmann, J. Colgate, and J. Patton. "Kinesthetic Interaction". In: *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005*. Chicago, IL, USA: IEEE, 2005, pp. 569–574. ISBN: 978-0-7803-9003-4. DOI: 10.1109/ICORR.2005.1502027. URL: <http://ieeexplore.ieee.org/document/1502027/>.
- [60] A. Özgür, S. Lemaignan, W. Johal, M. Beltran, M. Briod, L. Pereyre, F. Mondada, and P. Dillenbourg. "Cellulo: Versatile Handheld Robots for Education". In: *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. Vienna Austria: ACM, Mar. 2017, pp. 119–127. ISBN: 978-1-4503-4336-7. DOI: 10.1145/2909824.3020247. URL: <https://dl.acm.org/doi/10.1145/2909824.3020247>.
- [61] M.-F. Crainic and S. Preitl. "Ergonomic operating mode for a robot arm using a game-pad with two joysticks". In: *2015 IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics*. Timisoara, Romania: IEEE, May 2015, pp. 167–170. ISBN: 978-1-4799-9911-8. DOI: 10.1109/SACI.2015.7208192. URL: <http://ieeexplore.ieee.org/document/7208192/>.
- [62] C. Passenberg, A. Peer, and M. Buss. "A survey of environment-, operator-, and task-adapted controllers for teleoperation systems". In: *Mechatronics 20.7* (2010). Special Issue on Design and Control Methodologies in Telerobotics, pp. 787–801. ISSN: 0957-4158. DOI: <https://doi.org/10.1016/j.mechatronics.2010.04.005>. URL: <https://www.sciencedirect.com/science/article/pii/S0957415810000735>.
- [63] P. Simoens, M. Dragone, and A. Saffiotti. "The Internet of Robotic Things: A review of concept, added value and applications". In: *International Journal of Advanced Robotic Systems* 15.1 (Jan. 2018), p. 172988141875942. ISSN: 1729-8814, 1729-8814. DOI: 10.1177/1729881418759424. URL: <http://journals.sagepub.com/doi/10.1177/1729881418759424>.
- [64] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper. "USARSim: a robot simulator for research and education". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. ISSN: 1050-4729. Rome, Italy: IEEE, Apr. 2007, pp. 1400–1405. ISBN: 978-1-4244-0602-9 978-1-4244-0601-2. DOI: 10.1109/ROBOT.2007.363180. URL: <http://ieeexplore.ieee.org/document/4209284/>.
- [65] R. Suzuki, A. Karim, T. Xia, H. Hedayati, and N. Marquardt. "Augmented Reality and Robotics: A Survey and Taxonomy for AR-enhanced Human-Robot Interaction and Robotic Interfaces". In: *CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–33. ISBN: 978-1-4503-9157-3. DOI: 10.1145/3491102.3517719. URL: <https://dl.acm.org/doi/10.1145/3491102.3517719>.
- [66] G. Adamides, G. Christou, C. Katsanos, M. Xenos, and T. Hadzilacos. "Usability Guidelines for the Design of Robot Teleoperation: A Taxonomy". In: *IEEE Transactions on Human-Machine Systems* 45.2 (Apr. 2015), pp. 256–262. ISSN: 2168-2291, 2168-2305. DOI: 10.1109/THMS.2014.2371048. URL: <http://ieeexplore.ieee.org/document/6977975/>.

- [67] M. E. Walker, H. Hedayati, and D. Szafir. "Robot Teleoperation with Augmented Reality Virtual Surrogates". In: *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Daegu, Korea (South): IEEE, Mar. 2019, pp. 202–210. ISBN: 978-1-5386-8555-6. DOI: 10.1109/HRI.2019.8673306. URL: <https://ieeexplore.ieee.org/document/8673306/>.
- [68] J. Y. C. Chen, E. C. Haas, and M. J. Barnes. "Human Performance Issues and User Interface Design for Teleoperated Robots". In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37.6 (Nov. 2007), pp. 1231–1245. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2007.905819. URL: <http://ieeexplore.ieee.org/document/4343985/>.
- [69] A. B. Oving and J. B. van Erp. "Driving with a Head-Slaved Camera System". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 45.18 (Oct. 2001), pp. 1372–1376. ISSN: 2169-5067, 1071-1813. DOI: 10.1177/154193120104501812. URL: <http://journals.sagepub.com/doi/10.1177/154193120104501812>.
- [70] A. Kanduri, G. Thomas, N. Cabrol, E. Grin, and R. Anderson. "The (In)Accuracy of Novice Rover Operators' Perception of Obstacle Height From Monoscopic Images". In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 35.4 (July 2005), pp. 505–512. ISSN: 1083-4427. DOI: 10.1109/TSMCA.2005.850601. URL: <http://ieeexplore.ieee.org/document/1453698/>.
- [71] J. Carlson and R. Murphy. "How UGVs physically fail in the field". In: *IEEE Transactions on Robotics* 21.3 (June 2005), pp. 423–437. ISSN: 1552-3098. DOI: 10.1109/TR0.2004.838027. URL: <http://ieeexplore.ieee.org/document/1435486/>.
- [72] D. R. Lampton, D. P. McDonald, M. Singer, and J. P. Bliss. "Distance Estimation in Virtual Environments". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 39.20 (Oct. 1995), pp. 1268–1272. ISSN: 2169-5067, 1071-1813. DOI: 10.1177/154193129503902006. URL: <http://journals.sagepub.com/doi/10.1177/154193129503902006>.
- [73] S. Hughes and M. Lewis. "Task-Driven Camera Operations for Robotic Exploration". In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 35.4 (July 2005), pp. 513–522. ISSN: 1083-4427. DOI: 10.1109/TSMCA.2005.850602. URL: <http://ieeexplore.ieee.org/document/1453699/>.
- [74] R. Murphy. "Human–Robot Interaction in Rescue Robotics". In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 34.2 (May 2004), pp. 138–153. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2004.826267. URL: <http://ieeexplore.ieee.org/document/1291662/>.
- [75] D. Stewart. "A Platform with Six Degrees of Freedom". In: *Proceedings of the Institution of Mechanical Engineers* 180.1 (1965), pp. 371–386. DOI: 10.1243/PIME_PROC_1965_180_029_02. eprint: https://doi.org/10.1243/PIME_PROC_1965_180_029_02. URL: https://doi.org/10.1243/PIME_PROC_1965_180_029_02.

- [76] S. Bier, R. Li, and W. Wang. "A Full-Dimensional Robot Teleoperation Platform". In: *2020 11th International Conference on Mechanical and Aerospace Engineering (ICMAE)*. Athens, Greece: IEEE, July 2020, pp. 186–191. ISBN: 978-1-72818-322-0. DOI: 10.1109/ICMAE50897.2020.9178871. URL: <https://ieeexplore.ieee.org/document/9178871/>.
- [77] M. Hirschmanner, C. Tsiourti, T. Patten, and M. Vincze. "Virtual Reality Teleoperation of a Humanoid Robot Using Markerless Human Upper Body Pose Imitation". In: *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. Toronto, ON, Canada: IEEE, Oct. 2019, pp. 259–265. ISBN: 978-1-5386-7630-1. DOI: 10.1109/Humanoid s43949.2019.9035064. URL: <https://ieeexplore.ieee.org/document/9035064/>.
- [78] Seung Keun Cho, Hong Zhe Jin, Jang Myung Lee, and Bin Yao. "Teleoperation of a Mobile Robot Using a Force-Reflection Joystick With Sensing Mechanism of Rotating Magnetic Field". In: *IEEE/ASME Transactions on Mechatronics* 15.1 (Feb. 2010), pp. 17–26. ISSN: 1083-4435, 1941-014X. DOI: 10.1109/TMECH.2009.2013848. URL: <http://ieeexplore.ieee.org/document/4797870/>.
- [79] P. Chotiprayanakul, D. Wang, N. Kwok, and D. Liu. "A HAPTIC BASE HUMAN ROBOT INTERACTION APPROACH FOR ROBOTIC GRIT BLASTING". In: ().
- [80] I. S. Cardenas, K. A. Vitullo, M. Park, J.-H. Kim, M. Benitez, C. Chen, and L. Ohrn-McDaniels. "Telesuit: An Immersive User-Centric Telepresence Control Suit". In: *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Daegu, Korea (South): IEEE, Mar. 2019, pp. 654–655. ISBN: 978-1-5386-8555-6. DOI: 10.1109/HRI.2019.8673228. URL: <https://ieeexplore.ieee.org/document/8673228/>.
- [81] S. J. Jorgensen, M. Wonsick, M. Paterson, A. Watson, I. Chase, and J. S. Mehling. "Cockpit Interface for Locomotion and Manipulation Control of the NASA Valkyrie Humanoid in Virtual Reality (VR)". In: ().
- [82] O. Liu, D. Rakita, B. Mutlu, and M. Gleicher. "Understanding human-robot interaction in virtual reality". In: *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. Lisbon: IEEE, Aug. 2017, pp. 751–757. ISBN: 978-1-5386-3518-6. DOI: 10.1109/ROMAN.2017.8172387. URL: <http://ieeexplore.ieee.org/document/8172387/>.
- [83] X. Tang and H. Yamada. "Tele-operation Construction Robot Control System with Virtual Reality Technology". In: *Procedia Engineering* 15 (2011), pp. 1071–1076. ISSN: 18777058. DOI: 10.1016/j.proeng.2011.08.198. URL: <https://linkinghub.elsevier.com/retrieve/pii/S1877705811016997>.
- [84] R. Pausch, D. Proffitt, and G. Williams. "Quantifying immersion in virtual reality". In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*. USA: ACM Press, 1997, pp. 13–18. ISBN: 978-0-89791-896-1. DOI: 10.1145/258734.258744. URL: <http://portal.acm.org/citation.cfm?doid=258734.258744>.

- [85] A. Kulshreshth and J. J. LaViola. "Evaluating performance benefits of head tracking in modern video games". In: *Proceedings of the 1st symposium on Spatial user interaction*. Los Angeles California USA: ACM, July 2013, pp. 53–60. ISBN: 978-1-4503-2141-9. DOI: 10.1145/2491367.2491376. URL: <https://dl.acm.org/doi/10.1145/2491367.2491376>.
- [86] S. J. Jorgensen, M. W. Lanighan, S. S. Bertrand, A. Watson, J. S. Altemus, R. S. Askew, L. Bridgwater, B. Domingue, C. Kendrick, J. Lee, M. Paterson, J. Sanchez, P. Beeson, S. Gee, S. Hart, A. H. Quispe, R. Griffin, I. Lee, S. McCrory, L. Sentis, J. Pratt, and J. S. Mehling. "Deploying the NASA Valkyrie Humanoid for IED Response: An Initial Approach and Evaluation Summary". In: *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. Toronto, ON, Canada: IEEE, Oct. 2019, pp. 1–8. ISBN: 978-1-5386-7630-1. DOI: 10.1109/Humanoids43949.2019.9034993. URL: <https://ieeexplore.ieee.org/document/9034993/>.
- [87] P. Milgram and F. Kishino. "A Taxonomy of Mixed Reality Visual Displays". In: *IEICE Transactions on Information Systems* (Dec. 1994).
- [88] O. Hugues, P. Fuchs, and O. Nannipieri. "New Augmented Reality Taxonomy: Technologies and Features of Augmented Environment". In: *Handbook of Augmented Reality*. Ed. by B. Furht. New York, NY: Springer New York, 2011, pp. 47–63. ISBN: 978-1-4614-0063-9 978-1-4614-0064-6. DOI: 10.1007/978-1-4614-0064-6_2. URL: https://link.springer.com/10.1007/978-1-4614-0064-6_2.
- [89] S. A. Green, J. G. Chase, X. Chen, and M. Billingham. "Evaluating the augmented reality human-robot collaboration system". In: *International Journal of Intelligent Systems Technologies and Applications* 8.1/2/3/4 (2010), p. 130. ISSN: 1740-8865, 1740-8873. DOI: 10.1504/IJISTA.2010.030195. URL: <http://www.inderscience.com/link.php?id=30195>.
- [90] Microsoft. *Hololens*. URL: <https://www.microsoft.com/en-us/hololens>.
- [91] J. Y. Oh, J.-H. Park, and J.-M. Park. "FingerTouch: Touch Interaction Using a Fingernail-Mounted Sensor on a Head-Mounted Display for Augmented Reality". In: *IEEE Access* 8 (2020), pp. 101192–101208. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2997972. URL: <https://ieeexplore.ieee.org/document/9102253/>.
- [92] I. S. Cardenas, K. Powlison, and J.-H. Kim. "Reducing Cognitive Workload in Telepresence Lunar - Martian Environments Through Audiovisual Feedback in Augmented Reality". In: *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. Boulder CO USA: ACM, Mar. 2021, pp. 463–466. ISBN: 978-1-4503-8290-8. DOI: 10.1145/3434074.3447214. URL: <https://dl.acm.org/doi/10.1145/3434074.3447214>.
- [93] A. Tang, C. Owen, F. Biocca, and W. Mou. "Comparative Effectiveness of Augmented Reality in Object Assembly". In: *NEW HORIZONS* 5 (2003).
- [94] M. Leap. *Magic Leap 2*. URL: <https://www.magicleap.com/magic-leap-2>.

- [95] Y.-C. Tung, C.-Y. Hsu, H.-Y. Wang, S. Chyou, J.-W. Lin, P.-J. Wu, A. Valstar, and M. Y. Chen. "User-Defined Game Input for Smart Glasses in Public Space". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. Seoul Republic of Korea: ACM, Apr. 2015, pp. 3327–3336. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702214. URL: <https://dl.acm.org/doi/10.1145/2702123.2702214>.
- [96] A. K. Orphanides and C. S. Nam. "Touchscreen interfaces in context: A systematic review of research into touchscreens across settings, populations, and implementations". In: *Applied Ergonomics* 61 (May 2017), pp. 116–143. ISSN: 00036870. DOI: 10.1016/j.apergo.2017.01.013. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0003687017300212>.
- [97] E. A. Johnson. "Touch Display – A novel input/output device for computers". In: *Electronics Letter* 1 (1965), pp. 219–220.
- [98] H. Kang and G. Shin. "Hand usage pattern and upper body discomfort of desktop touchscreen users". In: *Ergonomics* 57.9 (Sept. 2014), pp. 1397–1404. ISSN: 0014-0139, 1366-5847. DOI: 10.1080/00140139.2014.924574. URL: <http://www.tandfonline.com/doi/abs/10.1080/00140139.2014.924574>.
- [99] G. Shin and X. Zhu. "User discomfort, work posture and muscle activity while using a touchscreen in a desktop PC setting". In: *Ergonomics* 54.8 (Aug. 2011), pp. 733–744. ISSN: 0014-0139, 1366-5847. DOI: 10.1080/00140139.2011.592604. URL: <https://www.tandfonline.com/doi/full/10.1080/00140139.2011.592604>.
- [100] J. G. Young, M. Trudeau, D. Odell, K. Marinelli, and J. T. Dennerlein. "Touch-screen tablet user configurations and case-supported tilt affect head and neck flexion angles". In: *Work* 41.1 (2012), pp. 81–91. ISSN: 10519815. DOI: 10.3233/WOR-2012-1337. URL: <https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/WOR-2012-1337>.
- [101] K. A. Siek, Y. Rogers, and K. H. Connelly. "Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs". In: *Human-Computer Interaction - INTERACT 2005*. Ed. by M. F. Costabile and F. Paternò. Vol. 3585. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 267–280. ISBN: 978-3-540-28943-2 978-3-540-31722-7. DOI: 10.1007/11555261_24. URL: http://link.springer.com/10.1007/11555261_24.
- [102] J. Nadvornik and P. Smutny. "Remote control robot using Android mobile device". In: *Proceedings of the 2014 15th International Carpathian Control Conference (ICCC)*. Velke Karlovice, Czech Republic: IEEE, May 2014, pp. 373–378. ISBN: 978-1-4799-3528-4 978-1-4799-3527-7. DOI: 10.1109/CarpathianCC.2014.6843630. URL: <http://ieeexplore.ieee.org/document/6843630/>.
- [103] Young-Hoon Jeon and Hyunsik Ahn. "A multimodal ubiquitous interface system using smart phone for human-robot interaction". In: *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. Incheon: IEEE, Nov. 2011, pp. 764–

767. ISBN: 978-1-4577-0723-0 978-1-4577-0722-3 978-1-4577-0721-6. DOI: 10.1109/URAI.2011.6146007. URL: <http://ieeexplore.ieee.org/document/6146007/>.
- [104] A. Sekmen, A. Koku, and S. Zein-Sabatto. "Human robot interaction via cellular phones". In: *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*. Vol. 4. Washington, DC, USA: IEEE, 2003, pp. 3937–3942. ISBN: 978-0-7803-7952-7. DOI: 10.1109/ICSMC.2003.1244503. URL: <http://ieeexplore.ieee.org/document/1244503/>.
- [105] L. Liu, X. Zhu, and Y. Tang. "Indoor surveillance robot controlled by a smart phone". In: *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Shenzhen, China: IEEE, Dec. 2013, pp. 1875–1880. ISBN: 978-1-4799-2744-9. DOI: 10.1109/ROBIO.2013.6739741. URL: <http://ieeexplore.ieee.org/document/6739741/>.
- [106] T. Zafar, M. Khan, A. Nawaz, and K. Ahmad. "Smart phone interface for robust control of mobile robots". In: *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. Espinho, Portugal: IEEE, May 2014, pp. 42–46. ISBN: 978-1-4799-4254-1. DOI: 10.1109/ICARSC.2014.6849760. URL: <http://ieeexplore.ieee.org/document/6849760/>.
- [107] R. Luo, Tse Min Chen, and Chih-Chen Yih. "Intelligent autonomous mobile robot control through the Internet". In: *ISIE'2000. Proceedings of the 2000 IEEE International Symposium on Industrial Electronics (Cat. No.00TH8543)*. Vol. 1. Cholula, Puebla, Mexico: IEEE, 2000, PL6–P11. ISBN: 978-0-7803-6606-0. DOI: 10.1109/ISIE.2000.930473. URL: <http://ieeexplore.ieee.org/document/930473/>.
- [108] G.-P. Liu, Y. Xia, D. Rees, and W. Hu. "Design and Stability Criteria of Networked Predictive Control Systems With Random Network Delay in the Feedback Channel". In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37.2 (Mar. 2007), pp. 173–184. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2006.886987. URL: <http://ieeexplore.ieee.org/document/4106032/>.
- [109] R. Luo, T. Lin, H. Chen, and K. Su. "Multisensor Based Security Robot System for Intelligent Building". In: *2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. Heidelberg, Germany: IEEE, Sept. 2006, pp. 408–413. ISBN: 978-1-4244-0567-1 978-1-4244-0566-4. DOI: 10.1109/MFI.2006.265589. URL: <http://ieeexplore.ieee.org/document/4042006/>.
- [110] Sung Wook Moon, Young Jin Kim, Ho Jun Myeong, Chang Soo Kim, Nam Ju Cha, and Dong Hwan Kim. "Implementation of smartphone environment remote control and monitoring system for Android operating system-based robot platform". In: *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. Incheon: IEEE, Nov. 2011, pp. 211–214. ISBN: 978-1-4577-0723-0 978-1-4577-0722-3 978-1-4577-0721-6. DOI: 10.1109/URAI.2011.6145963. URL: <http://ieeexplore.ieee.org/document/6145963/>.

- [111] M. Selvam. "SMART PHONE BASED ROBOTIC CONTROL FOR SURVEILLANCE APPLICATIONS". In: *International Journal of Research in Engineering and Technology* 03.03 (Mar. 2014), pp. 229–232. ISSN: 23217308, 23191163. DOI: 10.15623/ijret.2014.0303043. URL: <https://ijret.org/volumes/2014v03/i03/IJRET20140303043.pdf>.
- [112] M. Jiang, Y. Zhao, G. Liu, C. Liu, L. Zhu, and J. Sun. "The design of remote terminal for searching radiation source robot". In: *2017 Chinese Automation Congress (CAC)*. 2017, pp. 793–796. DOI: 10.1109/CAC.2017.8242874.
- [113] G. Murthy and R. Jadon. "A review of vision based hand gestures recognition". In: *International Journal of Information Technology and Knowledge Management 2.2* (2009), pp. 405–410.
- [114] M. Schroder, C. Elbrechter, J. Maycock, R. Haschke, M. Botsch, and H. Ritter. "Real-time hand tracking with a color glove for the actuation of anthropomorphic robot hands". In: *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. Osaka, Japan: IEEE, Nov. 2012, pp. 262–269. ISBN: 978-1-4673-1369-8. DOI: 10.1109/HUMANOIDS.2012.6651530. URL: <http://ieeexplore.ieee.org/document/6651530/>.
- [115] T. Gump, P. Azad, K. Welke, E. Oztop, R. Dillmann, and G. Cheng. "Unconstrained Real-time Markerless Hand Tracking for Humanoid Interaction". In: *2006 6th IEEE-RAS International Conference on Humanoid Robots*. University of Genova, Genova, Italy: IEEE, Dec. 2006, pp. 88–93. ISBN: 978-1-4244-0199-4 978-1-4244-0200-7. DOI: 10.1109/ICHR.2006.321368. URL: <http://ieeexplore.ieee.org/document/4115585/>.
- [116] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan. *3D Hand Shape and Pose Estimation from a Single RGB Image*. arXiv:1903.00812 [cs]. Apr. 2019. URL: <http://arxiv.org/abs/1903.00812>.
- [117] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann. *MediaPipe Hands: On-device Real-time Hand Tracking*. arXiv:2006.10214 [cs]. June 2020. URL: <http://arxiv.org/abs/2006.10214>.
- [118] S. Sreenath, D. I. Daniels, A. S. D. Ganesh, Y. S. Kuruganti, and R. G. Chittawadigi. "Monocular Tracking of Human Hand on a Smart Phone Camera using MediaPipe and its Application in Robotics". In: *2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC)*. Bangalore, India: IEEE, Sept. 2021, pp. 1–6. ISBN: 978-1-66543-240-5. DOI: 10.1109/R10-HTC53172.2021.9641542. URL: <https://ieeexplore.ieee.org/document/9641542/>.
- [119] J.-N. Voigt-Antons, T. Kojic, D. Ali, and S. Moller. "Influence of Hand Tracking as a Way of Interaction in Virtual Reality on User Experience". In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. Athlone, Ireland: IEEE, May 2020, pp. 1–4. ISBN: 978-1-72815-965-2. DOI: 10.1109/QoMEX48832.2020.9123085. URL: <https://ieeexplore.ieee.org/document/9123085/>.

- [120] T. Mergner, M. Funk, and V. Lippi. "Embodiment and Humanoid Robotics". In: *Philosophisches Handbuch Künstliche Intelligenz*. Ed. by K. Mainzer. Series Title: Springer Reference Geisteswissenschaften. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 1–27. ISBN: 978-3-658-23715-8. DOI: 10.1007/978-3-658-23715-8_23-1. URL: http://link.springer.com/10.1007/978-3-658-23715-8_23-1.
- [121] B. Miller and D. Feil-Seifer. "Embodiment, Situatedness, and Morphology for Humanoid Robots Interacting with People". In: *Humanoid Robotics: A Reference*. Ed. by A. Goswami and P. Vadakkepat. Dordrecht: Springer Netherlands, 2017, pp. 1–23. ISBN: 978-94-007-7194-9. DOI: 10.1007/978-94-007-7194-9_130-1. URL: http://link.springer.com/10.1007/978-94-007-7194-9_130-1.
- [122] A. Cangelosi and T. Ogata. "Speech and Language in Humanoid Robots". In: *Humanoid Robotics*. Publisher Copyright: © Springer Nature B.V. 2019. Springer Netherlands, Jan. 2018, pp. 2261–2292. ISBN: 9789400760455. DOI: 10.1007/978-94-007-6046-2_135.
- [123] K. Lohan, H. Lehmann, C. Dondrup, F. Broz, and H. Kose. "Enriching the human-robot interaction loop with natural, semantic and symbolic gestures". In: *Humanoid Robotics*. Ed. by A. Goswami and P. Vadakkepat. Springer, Sept. 2017, pp. 1–21. DOI: 10.1007/978-94-007-7194-9_136-1.
- [124] C. Scheier and R. Pfeifer. "The Embodied Cognitive Science Approach". In: *Dynamics, Synergetics, Autonomous Agents*, pp. 159–179. DOI: 10.1142/9789812815354_0011. eprint: https://www.worldscientific.com/doi/pdf/10.1142/9789812815354_0011. URL: https://www.worldscientific.com/doi/abs/10.1142/9789812815354_0011.
- [125] A. Haans and W. A. IJsselsteijn. "Embodiment and telepresence: Toward a comprehensive theoretical framework". In: *Interacting with Computers* 24.4 (2012), pp. 211–218. DOI: 10.1016/j.intcom.2012.04.010.
- [126] M. Blow, K. Dautenhahn, A. Appleby, C. Nehaniv, and D. Lee. "Perception of Robot Smiles and Dimensions for Human-Robot Interaction Design". In: *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*. Univ. of Hertfordshire, Hatfield, UK: IEEE, Sept. 2006, pp. 469–474. ISBN: 978-1-4244-0564-0 978-1-4244-0565-7. DOI: 10.1109/ROMAN.2006.314372. URL: <http://ieeexplore.ieee.org/document/4107851/>.
- [127] A. Prakash and W. A. Rogers. "Why Some Humanoid Faces Are Perceived More Positively Than Others: Effects of Human-Likeness and Task". In: *International Journal of Social Robotics* 7.2 (Apr. 2015), pp. 309–331. ISSN: 1875-4791, 1875-4805. DOI: 10.1007/s12369-014-0269-4. URL: <http://link.springer.com/10.1007/s12369-014-0269-4>.
- [128] M. Stapleton. "Steps to a "Properly Embodied" cognitive science". In: *Cognitive Systems Research* 22-23 (2013), pp. 1–11. ISSN: 1389-0417. DOI: <https://doi.org/10.1016/j.cogsys.2012.05.001>. URL: <https://www.sciencedirect.com/science/article/pii/S1389041712000241>.

- [129] J. Złotowski, E. Strasser, and C. Bartneck. “Dimensions of anthropomorphism: from humanness to humanlikeness”. In: *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. Bielefeld Germany: ACM, Mar. 2014, pp. 66–73. ISBN: 978-1-4503-2658-2. DOI: 10.1145/2559636.2559679. URL: <https://dl.acm.org/doi/10.1145/2559636.2559679>.
- [130] J. Bongard. “The Utility of Evolving Simulated Robot Morphology Increases with Task Complexity for Object Manipulation”. In: *Artificial Life* 16.3 (July 2010). _eprint: <https://direct.mit.edu/artl/article-pdf/16/3/201/1662657/artl.2010.bongard.024.pdf>, pp. 201–223. ISSN: 1064-5462. DOI: 10.1162/artl.2010.Bongard.024. URL: <https://doi.org/10.1162/artl.2010.Bongard.024>.
- [131] M. Gonzalez-Franco, D. Perez-Marcos, B. Spanlang, and M. Slater. “The contribution of real-time mirror reflections of motor actions on virtual body ownership in an immersive virtual environment”. In: *2010 IEEE Virtual Reality Conference (VR)*. Waltham, MA: IEEE, Mar. 2010, pp. 111–114. ISBN: 978-1-4244-6237-7 978-1-4244-6238-4. DOI: 10.1109/VR.2010.5444805. URL: <http://ieeexplore.ieee.org/document/5444805/>.
- [132] M. Botvinick and J. Cohen. “Rubber hands ‘feel’ touch that eyes see”. In: *Nature* 391.6669 (Feb. 1998), pp. 756–756. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/35784. URL: <https://www.nature.com/articles/35784>.
- [133] B. Lenggenhager, T. Tadi, T. Metzinger, and O. Blanke. “Video Ergo Sum: Manipulating Bodily Self-Consciousness”. In: *Science* 317.5841 (Aug. 2007), pp. 1096–1099. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1143439. URL: <https://www.science.org/doi/10.1126/science.1143439>.
- [134] H. Hapuarachchi, H. Ishimoto, M. Sugimoto, M. Inami, and M. Kitazaki. “Embodiment of an Avatar with Unnatural Arm Movements”. In: *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. Singapore, Singapore: IEEE, Oct. 2022, pp. 772–773. ISBN: 978-1-66545-365-3. DOI: 10.1109/ISMAR-Adjunct57072.2022.00163. URL: <https://ieeexplore.ieee.org/document/9974276/>.
- [135] M. Lombard and T. Ditton. “At the Heart of It All: The Concept of Presence”. In: *Journal of Computer-Mediated Communication* 3.2 (Sept. 1997). ISSN: 1083-6101. DOI: 10.1111/j.1083-6101.1997.tb00072.x. URL: <https://doi.org/10.1111/j.1083-6101.1997.tb00072.x>.
- [136] J. H. Murray. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. USA: The Free Press, 1997. ISBN: 0684827239.
- [137] N. C. Nilsson, R. Nordahl, and S. Serafin. “Immersion Revisited: A review of existing definitions of immersion and their relation to different theories of presence”. In: *Human Technology* 12.2 (Nov. 2016), pp. 108–134. ISSN: 17956889. DOI: 10.17011/ht/urn.201611174652. URL: <http://humantechnology.jyu.fi/archive/vol-12/issue-2/immersion-revisited>.
- [138] L. Ermi and F. Mäyrä. “Fundamental Components of the Gameplay Experience: Analysing Immersion.” In: Jan. 2005.

- [139] A. McMAHAN. “Immersion, Engagement, and Presence”. In: (2003).
- [140] E. Brown and P. Cairns. “A grounded investigation of game immersion”. In: *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. Vienna Austria: ACM, Apr. 2004, pp. 1297–1300. ISBN: 978-1-58113-703-3. DOI: 10.1145/985921.986048. URL: <https://dl.acm.org/doi/10.1145/985921.986048>.
- [141] M. Slater. “Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 364.1535 (Dec. 2009), pp. 3549–3557. ISSN: 0962-8436, 1471-2970. DOI: 10.1098/rstb.2009.0138. URL: <https://royalsocietypublishing.org/doi/10.1098/rstb.2009.0138>.
- [142] M. Minsky. “Telepresence”. 1980.
- [143] M. Slater. “A Note on Presence Terminology”. In: *Presence Connect* 3 (Jan. 2003).
- [144] M. Slater and S. Wilbur. “A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments”. In: *Presence: Teleoperators and Virtual Environments* 6.6 (Dec. 1997), pp. 603–616. ISSN: 1054-7460. DOI: 10.1162/pres.1997.6.6.603. URL: <https://direct.mit.edu/pvar/article/6/6/603-616/18157>.
- [145] J. K. O’Regan and A. Noë. “A sensorimotor account of vision and visual consciousness”. In: *Behavioral and Brain Sciences* 24.5 (Oct. 2001), pp. 939–973. ISSN: 0140-525X, 1469-1825. DOI: 10.1017/S0140525X01000115. URL: https://www.cambridge.org/core/product/identifier/S0140525X01000115/type/journal_article.
- [146] T. B. Sheridan. “Musings on Telepresence and Virtual Presence”. In: *Papers from SRI’s 1991 Conference on Virtual Reality on Virtual Worlds: Real Challenges: Real Challenges*. SRI '91. Menlo Park, California, USA: Meckler Corporation, 1992, pp. 55–65. ISBN: 0887368700.
- [147] T. B. Sheridan. “Further Musings on the Psychophysics of Presence”. In: *Presence: Teleoperators and Virtual Environments* 5.2 (Aug. 1996), pp. 241–246. DOI: 10.1162/pres.1996.5.2.241. URL: <https://doi.org/10.1162/pres.1996.5.2.241>.
- [148] C. H. M. Kawabata. “Mixed Reality in Healthcare: Immersive Humanoid Robot Control for Patient Care”. Bachelor’s Thesis. Technische Universität München, 2022.
- [149] A. Gao. “An Augmented Reality based Humanoid Robot Teleoperation System for Medical Care”. Bachelor’s Thesis. Technische Universität München, 2023.
- [150] J. Lee and T. Kunii. “Constraint-based hand animation. Models and Techniques in Computer Animation”. In: *IEEE Computer Graphics and Application* (1993).
- [151] J. Kuch and T. Huang. “Human computer interaction via the human hand: a hand model”. In: *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*. Vol. 2. Pacific Grove, CA, USA: IEEE Comput. Soc. Press, 1994, pp. 1252–1256. ISBN: 978-0-8186-6405-2. DOI: 10.1109/ACSSC.1994.471659. URL: <http://ieeexplore.ieee.org/document/471659/>.
- [152] J. Brooke. “SUS - A quick and dirty usability scale”. In: ().

- [153] T. C. Peck and M. Gonzalez-Franco. “Avatar Embodiment. A Standardized Questionnaire”. In: *Frontiers in Virtual Reality* 1 (Feb. 2021), p. 575943. ISSN: 2673-4192. DOI: 10.3389/frvir.2020.575943. URL: <https://www.frontiersin.org/articles/10.3389/frvir.2020.575943/full>.
- [154] M. L. Fiedler, E. Wolf, N. Döllinger, M. Botsch, M. E. Latoschik, and C. Wienrich. “Embodiment and Personalization for Self-Identification with Virtual Humans”. In: *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. Shanghai, China: IEEE, Mar. 2023, pp. 799–800. ISBN: 9798350348392. DOI: 10.1109/VRW58643.2023.00242. URL: <https://ieeexplore.ieee.org/document/10108709/>.
- [155] M. Gonzalez-Franco and T. C. Peck. “Avatar Embodiment. Towards a Standardized Questionnaire”. In: *Frontiers in Robotics and AI* 5 (June 2018), p. 74. ISSN: 2296-9144. DOI: 10.3389/frobt.2018.00074. URL: <https://www.frontiersin.org/article/10.3389/frobt.2018.00074/full>.