Spatial Computing for Human-Robot Interaction in Cyber-Physical Systems

by

Yehor Karpichev
B.Sc., Tallinn University of Technology, 2022

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Mechanical Engineering

We acknowledge and respect the Lək̓ʷəŋən (Songhees and Xʷsepsəm/ Esquimalt) Peoples on whose territory the university stands, and the Lək̓ʷəŋən and W̱SÁNEĆ Peoples whose historical relationships with the land continue to this day.

Spatial Computing for Human-Robot Interaction in Cyber-Physical Systems

by

Yehor Karpichev
B.Sc., Tallinn University of Technology, 2022

Supervisory Committee

Dr. H. Najjaran, Supervisor
(Department of Mechanical Engineering)

Dr. B. Haworth, Outside Member
(Department of Computer Science)

ABSTRACT

This thesis investigates the challenges and opportunities in enhancing human-robot interaction (HRI) and collaboration by integrating spatial technologies and digital twin frameworks. It explores the technological transformations of Industry 4.0 and 5.0 in fostering closer human-machine collaboration, and examines the deployment challenges of robotic systems and their seamless integration into digital twin environments. A key focus is the role of immersive interfaces in enabling intuitive and effective human-robot communication. Contemporary spatial technologies increasingly blur the boundaries between physical and digital worlds, enhancing situational awareness and interaction fidelity, while promoting the human perspective within automated systems.

To support scalable deployment and modularity, a container-based pipeline was developed to encapsulate robotic systems and their digital twins, enabling real-time control, isolated execution, and seamless integration across heterogeneous software and hardware platforms. In parallel, the role of extended reality (XR) in facilitating human-robot communication was examined through both theoretical framing and practical implementation in virtual and mixed reality settings. The work also explores the integration of 3D Gaussian Splatting as an emerging technique for scene representation in immersive applications, including the HRI scenarios, to evaluate its potential for improving human understanding of the spatial context through highly realistic and natural environments.

Together, these contributions demonstrate a framework for human-centered robotics, leveraging spatial computing as a bridge between physical systems and virtual environments. By promoting more natural, intuitive, and context-aware interaction, this work lays the foundation for advancing human-centered cyber-physical systems. This is achieved by ensuring intelligent systems remain adaptable, transparent, and compatible with human interaction in real-world settings.

PREFACE

The work presented in this thesis has been done at the Advanced Control and Intelligent Systems Lab at the University of Victoria under supervision of Prof. Homayoun Najjaran. It has been an invaluable experience collaborating with colleagues from diverse backgrounds and interdisciplinary expertise, many of whom provided insights that supported approaching the subject from different perspectives and viewing problems in a new light. Several collaboratively written publications have contributed to writing this master's thesis.

- Inspired by the rise of automation and the concurrent demand for flexibility, the published work focused on theorizing how human-robot collaborative systems should operate, with a particular emphasis on the role of extended reality as enabling technology. In collaboration with co-authors, a human-in-the-loop framework was conceptualized, envisioning direct human involvement in the robot learning process to enhance adaptability and task generalization.

  [48] **Y. Karpichev**, T. Charter, J. Hong, A. M. Soufi Enayati, H. Honari, M. G. Tamizi, and H. Najjaran. Extended Reality for Enhanced Human-Robot Collaboration: A Human-in-the-Loop Approach. In 2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN), pages 1991–1998, 2024.

  Parts of this publication are mentioned in Chapters 2 and 4.

- The following co-authored paper presents a scalable ROS-Docker framework designed for digital twin applications across different hardware and software platforms. This work provided foundational tools that supported many of the experimental systems and integration workflows explored throughout this thesis.

  [49] **Y. Karpichev**, M. C. Zaouali, T. Charter, and H. Najjaran. A Deployable and Scalable ROS-Docker Framework for Multi-Platform Digital Twin Applications. TechRxiv, 2025.

  [Preprint; accepted for publication at the Canadian Conference on Electrical and Computer Engineering 2025]

  This work constitutes Chapter 3 of this thesis.

- Part of the content presented in Chapter 5 has been adopted in a following manuscript, focused on language-guided 3D Gaussian Splatting (3DGS). The paper surveys recent advances in 3DGS as a scene representation technique, particularly its integration with language and foundation models. The author's (of this thesis) contributions are primarily to the discussion of real-world applications, specifically in immersive technologies and asset generation.

  [121] M. C. Zaouali, T. Charter, **Y. Karpichev**, B. Haworth, and H. Najjjaran. A Study of the Framework and Real-World Applications of Language Embedding for 3D Scene Understanding. *arXiv*, 2025.

  [Preprint; submitted for potential publication]

Contents

Supervisory Committee **ii**

Abstract **iii**

Preface **iv**

Table of Contents **vi**

List of Tables **ix**

List of Figures **x**

Acknowledgements **xii**

Dedication **xiii**

1 Introduction **1**

1.1 Motivation 1

1.2 Objectives 2

1.3 Contributions 3

1.4 Thesis Outline 4

2 Background **6**

2.1 Fundamentals of Human-Robot Collaboration 6

2.1.1 Historical Context and Evolution 6

2.1.2 Types of Interaction Modalities 10

2.1.3 Interaction Levels in HRC 12

2.1.4 Metrics in HRI 13

2.2 Digital Twins 14

2.2.1 Definition 14

2.2.2 Characteristics and Applications 15
2.2.3 Digital Twins and Cyber-Physical Systems 18
2.3 Extended Reality in HRC 19
2.3.1 Immersive Technologies: Core Concepts 19
2.3.2 Real-World Industrial Applications of XR in HRC 20
2.3.3 Current Limitations and Emerging Trends 26

3 Containerization for Digital Twins **29**
3.1 Background 30
3.1.1 Robot Operating System 30
3.1.2 Virtual Machines vs Docker 30
3.2 Implementation 31
3.2.1 Containerized Environment 32
3.2.2 ROS Network 34
3.2.3 MoveIt Toolbox 34
3.2.4 Integration with Unity 35
3.2.5 Simulation Tools: RViz and Gazebo 36
3.3 Evaluation 37
3.4 Remarks and Future Development 40

4 Human-Centered Immersive Control for Robotics **41**
4.1 Human-in-the-Loop Framework Conceptualization 42
4.1.1 Manipulator Task Generalization 42
4.1.2 Human-in-the-Loop Component 44
4.1.3 Enabling Technologies 46
4.1.4 Remarks 47
4.2 Virtual Reality for Robot Teleoperation 48
4.2.1 Hardware Setup 48
4.2.2 Software Setup 48
4.2.3 Implemented Functionality 51
4.2.4 User Interface 52
4.2.5 Usability Considerations 53
4.3 From Virtual to Mixed Reality 55
4.3.1 Hardware and Software Setups 55
4.3.2 Experimental Design 57

4.4 Discussion: Interaction and Contextual Trade-offs between VR and MR in HRI 59

5 Realistic Scene Representation for Immersive Applications 61

5.1 Background 61
5.1.1 Novel View Synthesis Techniques 61
5.1.2 Facets of Realism in Virtual Environments 62
5.1.3 3DGS Integration with Game Engine and XR 63
5.2 Object-level Reconstruction: Preliminary Investigation 64
5.2.1 Case Study: Rubik's Cube 65
5.2.2 Visual Representation and Interactions 66
5.2.3 Quantitative Analysis and Metrics 68
5.3 Scene-level reconstruction: Research Lab 70
5.3.1 Installation and Setup 71
5.3.2 Reconstruction Analysis 71
5.3.3 Unity Integration and Robot Model Representation 73
5.4 Discussion 74

6 Concluding Remarks and Future Outlook 76

A Additional Information 79

A.1 Nielsen's Usability Heuristics 79
A.2 Head-Mounted Display Specifications 80
A.3 Formal Description of SOR Filtering and Dimensional Metrics 81
A.3.1 Statistical Outlier Removal with K-NN 81
A.3.2 Quantitative Evaluation Metrics 82

Bibliography 83

List of Tables

Table 2.1 Classification of Interaction Modalities in Human-Robot Collaboration. 12
Table 2.2 Industrial HRI levels. 13

Table 3.1 Evaluation Metrics for a Dockerized Robotic Application 38
Table 3.2 Resource Usage Summary for Docker Container 38
Table 3.3 Hardware and Software Specifications Used for Container Evaluation 40

Table 5.1 Axis-Aligned Size and Geometric Differences Between Original Object and Reconstructed Representations. 70

Table A.1 Nielsen's 10 Usability Heuristics for Interface Design 79
Table A.2 Technical Specifications of Meta Quest 2 and HoloLens 2 80

List of Figures

Figure 2.1 Industrial Revolutions Timeline 7
Figure 2.2 Conceptual Representation of a Digital Twin 16
Figure 2.3 Reality-Virtuality Continuum, adopted from [69] 19

Figure 3.1 Communication Pipeline for the Cyber-Physical System 32
Figure 3.2 Robot Control Using RViz: Motion Planning Is Handled by MoveIt, While the Point Cloud Captured by the Onboard Camera Is Visualized in the Scene 33
Figure 3.3 ROS Network Running in the Docker Container 35
Figure 3.4 The High-Level Architecture of the MoveIt `move_group` [80] . . 36
Figure 3.5 RViz-based Control of Kinova Gen 3 37
Figure 3.6 Gazebo Simulation of Kinova Gen 3 37

Figure 4.1 Proposed HRI level: *Fully Autonomous with the Human-in-the-Loop* 42
Figure 4.2 XR-Enabled Human-in-the-Loop Approach for Enhanced HRC 44
Figure 4.3 Meta Quest 2 with Touch Controllers 48
Figure 4.4 Virtual Replica of the ACIS Research Lab at the University of Victoria 50
Figure 4.5 Real-Time Robot Control via Virtual Reality. 52
Figure 4.6 UI for Manual Control of Manipulator Joints and Virtual Twin Preview. 53
Figure 4.7 Revised UI with Hand Tracking Implementation 54
Figure 4.8 Mixed Reality Headset – Microsoft Hololens 2 56
Figure 4.9 Hand Menu for the Robot Teleoperation in Mixed Reality . . . 57
Figure 4.10Marker Tracking and Robot Visualization in Mixed Reality . . 58

Figure 5.1 Overview of the Data Processing Pipeline: 3D Scanning, Followed by Point Cloud Refinement, and Game Engine Integration 66

Figure 5.2 Side Views of the Reconstructed Rubik's Cube Using Gaussian Splatting 66
Figure 5.3 Three Rubik's Cubes in the Virtual Reality Environment: Photogrammetry-Based (Left), CAD Model (Center), Gaussian Splat-Based (Right) 67
Figure 5.4 Interaction with Reconstructed Objects in Virtual and Mixed Realities 67
Figure 5.5 Zoomed-Out View of the Reconstructed Research Lab 72
Figure 5.6 Render Modes of Gaussian Splatting Reconstruction 73
Figure 5.7 Robot Representation within the 3DGS-based Environment in VR. 74

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Dr. Homayoun Najjaran, for his invaluable support and encouragement throughout my Master's program. I am especially thankful for the many opportunities and projects I had the pleasure of being part of under his mentorship. Working across areas such as robotics, extended reality, and machine learning has been a both challenging and rewarding experience.

I am grateful to my colleagues at the Advanced Control and Intelligent Systems Lab for their technical insights and collaborative spirit. In particular, I would like to thank my co-authors Amir M. S. Enayati, Homayoun Honari, Jayden Hong, Mehran G. Tamizi, and Todd Charter for their contributions and teamwork.

Also, I extend my heartfelt thanks to Alex Deaconu, Dina El-Kholy, and Olivia Margoto, with whom I had the privilege of working during my exchange at the UBC Okanagan as part of the NSERC CREATE in Immersive Technologies program. Your professionalism, enthusiasm, and friendship made the experience incredibly rewarding.

Another special thank-you goes to my close friends Ana Catarina Pereira, Aaron Chegini, Bernardo Leite, Diogo Bravo, and Victor Marrugat – you brought so much color and joy to everyday life, and your presence meant a lot throughout this journey.

A very special thanks to Mahmoud Chick Zaouali – more than just a colleague, you have been a true friend and mentor during times when I needed it most.

To Joel Sol – thank you for all the fun, the support, and the many Strava runs we shared.

Finally, I want to thank my parents for their endless encouragement and patience throughout this academic journey. Your emotional support over the past few years has meant the world to me.

DEDICATION

Scientific truth is beyond loyalty and disloyalty.

– *Foundation*, Isaac Asimov

Chapter 1

Introduction

This chapter presents the background that frames the motivation behind the thesis. It outlines the research questions and objectives that serve as guiding pillars throughout the study. The chapter concludes with a summary of the thesis structure and a brief description of the contributions made in each subsequent chapter.

1.1 Motivation

The rise of automation has provided an opportunity to achieve higher efficiency in manufacturing processes, yet it often compromises the flexibility required to promptly respond to evolving market needs and meet the demand for customization. Human-robot collaboration envisions to tackle these challenges by combining the strength and precision of machines with human ingenuity and perceptual understanding.

The evolution from Industry 4.0 to Industry 5.0 is a significant change in the manufacturing sector, from a focus on autonomous cyber-physical systems, IoT, and Big Data to more human-oriented approaches. Industry 5.0 is centered on humans working together with machines, combining robotic accuracy with human flexibility and intuition [45]. A key challenge herein is the creation of intuitive interfaces for communication that reflects the complexity of the manufacturing process while allowing for the adaptability and flexibility of users. Conventional robot software is typically centered on the performance of narrowly defined tasks, thereby allowing minimal scope for deviation. Even if machine learning translates to greater flexibility and improved decision-making ability, it should be recognized as an enabling tool and not the main means of communication between robots and humans.

1.2 Objectives

In today's rapidly evolving technological landscape, spatial and in-situ technologies are becoming increasingly accessible, more compact, and equipped with greater computational power. Immersive technologies, particularly when combined with advanced virtual representation techniques, are progressively blurring the boundaries between the physical and digital worlds, bringing us closer a new era of interconnected spatial environments. Understanding the key driving technologies, fundamental concepts, and design considerations behind these immersive systems is essential for improving their effectiveness, usability, and human acceptance. This work investigates these aspects within the context of human-robot collaboration, focusing on how immersive technologies can enhance interaction and control.

The following research questions guided the investigation and practical implementations presented in this thesis:

1. What impact do the technological innovations and societal changes of Industries 4.0 and 5.0 have on the evolution of human-robot communication?

2. What design principles ensure scalability, flexibility, and robustness of digital twin platforms across heterogeneous hardware and software environments?

3. How should extended reality (XR) frameworks be designed to enhance operator involvement and situational awareness in human-robot collaboration?

4. What are the benefits and limitations of virtual reality (VR) versus mixed reality (MR) in supporting operator's immersive robot control?

5. How does the representation of virtual environments influence user perception, decision-making, and interaction effectiveness?

In addressing the research objectives and questions, this work seeks to contribute meaningful insights into the evolving landscape of human-centric robotics and automation systems. The aim is not only to advance understanding of the enabling technologies but also to examine practical solutions that enhance cooperation between humans and robots – ultimately fostering the development of safer, more realistic, efficient, scalable, and adaptable systems.

1.3 Contributions

This thesis is designed to offer an in-depth examination of spatial technologies in the context of human-robot interaction (HRI). Each chapter builds upon the previous one, gradually revealing the complexity and interdisciplinary nature of the HRI domain.

Chapter 2 lays the theoretical groundwork by reviewing core concepts in human-robot collaboration, digital twins, and extended reality (XR). It provides historical context, tracing technological and societal shifts from early industrial revolutions to the human-centric vision of Industry 5.0, while discussing essential topics such as human-robot interaction levels, types of interaction modalities, and existent approaches to qualitative and quantitative benchmarks. The section on digital twins highlights their role in digitalization and their importance within Industry 4.0 and cyber-physical systems. Immersive technology is introduced as a key spatial computing technology, enabling immersive, intuitive interaction and situational awareness. Overall the chapter establishes the foundation for the technical work that follows and identifies the research gaps this thesis addresses.

Chapter 3 addresses the containerization of digital twin systems for robotics by leveraging Docker for the Robot Operating System (ROS). This approach creates a modular, scalable, and portable development environment that simplifies deployment and maintenance across different platforms. The chapter details the design and implementation of this containerized architecture, which forms the technical foundation for the robot teleoperation experiments presented in later chapters. By enabling consistent and reproducible setups, this work contributes to enhancing the flexibility and robustness of digital twin-based robotic applications.

Chapter 4 combines theoretical insights with practical implementation. It begins by proposing a conceptual extended reality (XR) framework aimed at enhancing operator involvement in the robot learning process. Building on this foundation, the chapter details the development of real-time user interaction methods within a robot teleoperation context, emphasizing a user-centered design approach. It compares virtual and mixed reality technologies, headsets, and interfaces, assessing their effectiveness for intuitive and responsive robot control. This chapter introduces a conceptual vision and a concrete implementation that collectively advance the integration of XR in human-robot collaboration.

Chapter 5, building on the immersive experiences developed in Chapter 4, explores advanced 3D scene representation techniques, with a particular focus on 3D

Gaussian Splatting (3DGS) as a novel and emerging method. This chapter evaluates the potential of 3DGS to generate high-fidelity, real-time renderings at both the object and scene levels. It further examines the implications of these representations for enhancing perceptual realism and interactive responsiveness in virtual environments, including within human-robot interaction scenarios. The chapter also presents implementation strategies for integrating 3DGS into game engines and XR platforms, supported by a series of proof-of-concept experiments that demonstrate both the feasibility and current limitations of the approach.

Together, these chapters form a coherent progression – from theoretical foundations and system infrastructure to immersive control and advanced interaction modalities, resulting in a progressive vision for intelligent human-robot interaction within immersive environments.

1.4 Thesis Outline

The document consists of six chapters in total: an introduction, a background chapter, three technical chapters that present both theoretical and implementation-based contributions, and a conclusion with future outlook. The following list provides an overview of the thesis structure and covered topics in each chapter.

- **Chapter 1:** Introduction
  - Introduces the motivation, research objectives, and scope of the thesis.
- **Chapter 2:** Background
  - This chapter presents a literature review on human-robot collaboration, covering fundamental interaction concepts, digital twins and their integration with cyber-physical systems, and the role of extended reality (XR) technologies as communication interface.
- **Chapter 3:** Containerization for Digital Twins
  - Describes the technical foundation and implementation of a containerized architecture for robotics digital twins, including the use of ROS, Docker, MoveIt, and simulation tools (Gazebo and RViz).
- **Chapter 4:** Human-Centered Immersive Control for Robotics

  - Introduces a conceptual framework for human-robot skill transfer enabled by extended reality.
  - Demonstrates immersive robot control through both virtual and mixed reality implementations, highlighting the unique considerations and development differences across varying levels of immersion.

- **Chapter 5:** Realistic Scene Representation for Immersive Applications
  - Investigates 3D Gaussian Splatting as a method for asset generation and scene representation in virtual environments. The chapter explores its integration with XR platforms and discusses its potential to enhance realism, spatial fidelity, and human-robot interaction within immersive settings.

- **Chapter 6:** Concluding Remarks and Future Outlook
  - Concludes the conducted research and findings, and provides an outlook for the future works.

Chapter 2

Background

This chapter provides the foundation for the research presented in this thesis. It offers contextual background and introduces key concepts relevant to the study of human-robot interaction (HRI) through the integration of digital twins and immersive technologies. While this chapter includes a relevant literature review, subsequent chapters (3, 4, and 5) also provide additional background specific to the technologies or use cases discussed therein.

2.1 Fundamentals of Human-Robot Collaboration

2.1.1 Historical Context and Evolution

Modern Industrial Revolutions: The Rise of Industry 4.0

Four major industrial revolutions (Figure 2.1), taking place in the last several centuries, each of them having a great impact on the manufacturing processes and societal organization. The first industrial revolution started in the late 18th century and was driven primarily by harnessing and utilizing the energy sources like coal and water. The steam engine is widely recognized as one of the main inventions of this period [117]. The second industrial revolution started in the early 1900s, building upon the foundations laid by the first with the establishment of new power sources like electricity and petroleum. This led to the creation of assembly lines and mass production. The third industrial revolution, in the 1960s, once again transformed the indsutry through the automation of processes enabled by the large scale integration of robotics and the information technologies [94]. Finally, Industry 4.0 emerged in the

early 2000s, driven by the spread of interconnected technologies that revolutionized traditional manufacturing and brought in the concept of smart manufacturing.

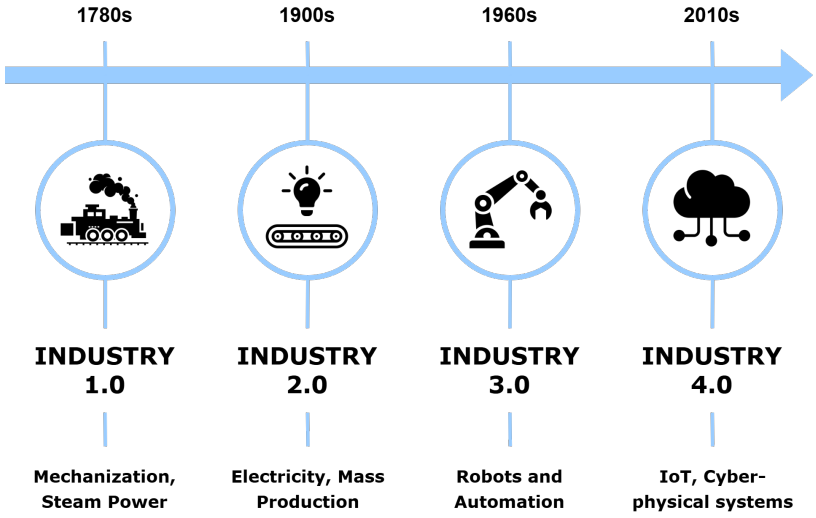


Figure 2.1: Industrial Revolutions Timeline

Industry 4.0 has not just brought in new technologies but also a new perspective to manufacturing processes, thereby blurring the gap between the virtual and physical worlds. Unlike in the past industrial revolutions, Industry 4.0 cannot be simply considered a linear continuation because of its unprecedented scope, velocity, and significance [117]. Key technologies driving this change include the Internet of Things (IoT), Artificial Intelligence (AI), Manufacturing Execution Systems (MES), Additive Manufacturing and rapid prototyping, sensors, cloud services, big data analytics, and simulations [20]. Together, these technologies enable the creation of cyber-physical systems – holistic systems that integrate physical components (like machines, robots, and sensors), computations (like algorithms, software, and control systems), and network connectivity (mainly through industrial IoT) [57]. These systems reflect physical processes using real-time data, effectively building a digital twin of the physical environment which can be analyzed, simulated, and optimized (more completely discussed in Section 2.2). The notion of the Smart Factory has arisen from this evolution, which is an intelligent manufacturing environment in which machines, devices, and systems are interconnected with autonomous decision-making ability [11]. Made possible by cyber-physical systems, smart factories use real-time data for optimizing, increasing resilience, and enhancing general manufacturing effectiveness.

Although these revolutions are commonly regarded as separate milestones, they

are in fact best viewed as a continuous sequence of events – one each drawing on the technological advances of its predecessor to develop increasingly complex forms of production [117].

Toward Industry 5.0: Human-Centric Approach

Building on the trajectory of the Industry 4.0, the emerging concept of Industry 5.0 shifts the focus from the technological advancement to sustainability and human-centricity, marking a potential turn toward value-driven and human-oriented innovation. In 2020, the European Commission attempted to define some of the characteristics of the Industry 5.0, characterizing the need in a shift of the technologies advancements and achievements of 4th industrial revolutions towards compatibility with the concept of circular economy, and adapting the human-centric strategy. In other words, promoting a shift in perspective from people serving the organizations, to organizations serving people.

However, Industry 5.0 on multiple levels is different from other industrial revolutions. Firstly, it must be stated that Industry 4.0 is still maturing and evolving, and numerous businesses around the globe are significantly behind on implementing the new technologies [32]. Secondly, all previous industrial revolutions were primarily technology-driven; meanwhile, Industry 5.0 is guided by the environmental and social factors. For example, Industry 4.0 achieves a high degree of automation and intelligence in manufacturing through technologies such as digital twins, the Internet of Things, and data analysis. While the Industry 5.0 builds on that foundation, emphasizing the human-centricity, resilience and sustainability as core values. It stresses on the human knowledge, flexibility, essentially ingenuity in the collaboration with the machines, therefore, promoting the sustainable and resilient economy, and value-driven innovation. Hence, as discussed by multiple researchers [119, 32], the Industry 5.0 shouldn't be considered as an independent revolution, but rather a policy framework that builds upon the advancements achieved by the Industry 4.0.

Focusing on the human-centric approaches, the Industry 5.0 involves integrating humans into virtual environments, introducing the concept of metaverses, where interactions with environmental digital assets are possible [119]. Achieving this requires technologies that allow for a more immersive and realistic representation of production processes. The industrial metaverse integrates technologies such as Digital Twins, Virtual Reality (VR), and the Industrial Internet of Things (IIoT) within

an interconnected framework to enable human-to-human and human-machine interactions in virtual environments [65]. Therefore, leading to the human-in-the-loop designs [36, 48] where the operators become an integral and vital part of the system. As such, the machines, the robots are seen as collaborative partners and assistants to the human operator, therefore, extending the human capabilities and promoting well-being [119]. Simultaneously, this results in a necessity of the development of the personalized and intuitive interfaces that can operate where the the physical and digital blends.

Emergence of Human-Robot Collaboration

The idea of the collaboration or interaction between humans and machines or robots is not new and has been mentioned throughout history. In a modern context, the emergence of the HRC field can be dedicated to the disruptive innovations in Industry 4.0 (and nowadays also 5.0), and paradigm shift from automation to collaboration [78]. In other words, recognizing that although robots may provide repeatability and precision, the humans' flexibility and world understanding can hardly be replaced [48]. This eventually led to the development of collaborative robots (cobots), collaborative environments, and formalized standards such as ISO10218. Today, most robotics manufacturers have introduced their own lines of cobots – well-known companies like Universal Robots, KUKA, Franka, ABB, Fanuc, and others – stressing on the compliance with safety standards (such as operational speed, force sensors) as well as on user-friendly programming. Furthermore, the recent advances in sensing, control algorithms, lightweight actuators, and machine learning have made it technically viable for robots to safely share spaces with humans [89].

Generally, there are numerous scenarios that may fall under the umbrella of the human-robot interaction (HRI). However, it started evolving as a separate and multi-disciplinary field in 1990s and early 2000s, so it may be considered a relatively new and rapidly changing [33]. The field brings together researchers from disciplines like robotics, computer science, psychology, artificial intelligence, and other. The multi-disciplinary collaboration is a good practice, and in fact a requirement in HRI research due to the high level of complexity and subjects overlaps. Some of the most common research directions within HRI are interaction interfaces, robot design, interaction modalities, task sharing, intention recognition, human factors and ergonomics, and learning from demonstration (LfD).

While HRI covers rather broadly any interaction between humans and robots, the human-robot collaboration (HRC) specifically refers to scenarios where humans and robots work together toward a shared goal, often in close proximity and including potentially overlapping roles [90]. The terms HRI and HRC, although are not the same, can be used interchangeably on multiple occasions. In case of HRC, some of the key areas of research include the design of collaborative workspaces that allow safe and efficient co-performance of tasks, and the distinction between task allocation and task sharing. Additionally, in any HRI context, increasing attention is given to trust and transparency in robotic systems – where both are crucial for effective interaction and user acceptance. This emphasizes once again the inherently multidisciplinary nature of the field.

2.1.2 Types of Interaction Modalities

As previous sections have introduced the increasing role of the human-robot collaboration field and the evolving need in the intuitive communication interfaces, it is also crucial to address the interaction modalities. Some of the more established methods are graphical user interfaces (GUIs), which offer visual elements such as panels and buttons, and joysticks, which provide direct, real-time physical control. These traditional tools remain widely used in industrial and teleoperation settings due to their reliability and ease of use. Besides those, there are numerous interaction modes (i.e. input channels) that exist for explicit and implicit ways of communicating. This section aims to offer a brief overview of the existent and known methods to complement the Chapter's review on the fundamental concepts in HRC. The summary is provided in Table 2.1.

Gestures seem like an intuitive way to communicate, as they are a natural mode of expression. Gestures can be classified as either contact-based or vision-based approaches [71]. The contact-based method requires gloves or similar sensors, which ultimately negates the naturalness for humans. An alternative is the vision-based method, where human gestures are detected and interpreted. Nonetheless, two major issues arise. Firstly, the gestures must be accurately detected and interpreted, requiring a reliable vision system. Secondly, gesture use can quickly become complex if each command requires a unique gesture; handling out-of-vocabulary gestures is also a challenge [12]. Furthermore, one of the most common issues in using gestures is the lack of unified standards among HRC researchers and available datasets.

Another interaction modality is voice commands. Natural language is likely the most intuitive for human operators, as it allows them to explicitly state commands and permits providing more complex contextual information when needed [124]. In industrial settings, voice commands offer robustness against lighting variations and shadows (in contrast to gesture vision-based systems). However, depending on the system configuration, it may be susceptible to background noise, and requires a robust language processing.

Gaze plays also a critical role in enhancing communication in HRC scenarios, providing information about the user's focus of attention, therefore, potential intentions and area or object of interests [111]. Head pose estimation often can be used as a substitute for gaze estimation since it is simpler in implementation, but also lacks precision. For example, the eye movements may indicate a shift in the operator's attention without any head motion. Studies have shown that systems utilizing eye gaze tracking achieve higher task efficiency, improved user satisfaction, and are perceived as more intelligent and natural [104]. Nonetheless, implementing the eye gaze tracking presents a number of challenges such as sensitivity to lightning, the need in the specialized equipment like high-resolution cameras, potential discomfort in case of the wearable devices, and difficulty in accurately detecting and interpreting eye features in the dynamic settings. However, gaze can be an intuitive and effective solution in HRI, especially when shared attention or precise coordination is required.

Finally, there are numerous other potential modes of communicating the user's intent implicitly. For example, that includes facial expressions that may indicate frustration, satisfaction, surprise, etc [104]. However, it is more commonly used in social robotics, often in the social interactions, user acceptability, and trust related studies. Body language, posture, and the user movements in the environment can be also analyzed and used as additional implicit cues in communication. For instance, leaning forward, stepping away, or operator's orientation relative to the robot counterpart can all be cues. Similarly, the user movements around the scene can be used by the robot for adjusting its path planning. Another, less common, yet emerging field and technology – physiological signals. Such as heart rate, stress levels (via wearable sensros), EEG waves, and other to infer cognitive load or intent [126].

Another important category related to interaction modalities is the physical guidance of the robot during task execution, commonly seen in kinesthetic teaching and Learning from Demonstration (LfD). In these methods, the human either manually moves the robot through the desired actions or performs the task for the robot to

observe and learn from [27]. While not strictly an interaction modality on its own, this approach plays a key role in conveying both knowledge and intent – making it especially valuable for non-experts and well-suited for training robots in complex or unstructured environments.

Table 2.1: Classification of Interaction Modalities in Human-Robot Collaboration.

Category	**Modality**	**Intent**
Direct Command Interfaces	Voice, Gesture, Joystick, Gaze	Explicit
Graphical Interfaces	GUIs (touch, XR panels, HUDs)	Explicit
Implicit Cues	Facial Expressions, Body Language, User Movements	Implicit
Teaching / Training	Kinesthetic Teaching, Learning from Demonstration (LfD), Motion Capture	N/A (task-level learning)

2.1.3 Interaction Levels in HRC

Working side by side with robots can be complex as both humans and robots bring different strengths, limitations, and roles to the table – and these can vary widely depending on the task and the environment. While many researchers have tried to define how these collaborations should work, most studies tend to focus on a specific case or some part of the interaction, while missing the bigger picture. To tackle this gap, Mukherjee et al. [71] proposed a more adaptable framework tailored to industrial settings. The proposed classification takes into account factors like the nature of the task being done, the layout of the workspace, the level of autonomy of the robot, and whether the agents come into the physical contact. The system organizes human-robot collaboration into six levels – from level zero (where the robot is completely pre-programmed) to level five (where the robot is fully autonomous powered by the machine learning algorithms). Interestingly, at both ends of the spectrum, there's actually no human involvement. The real collaboration happens at level four, where humans and robots actively work together toward shared goals. A more structured summary of this framework can be found in Table 2.2.

Based on this classification, a key insight emerges: effective human-robot collaboration depends less on the robot itself and more on the design of the overall application

system. Furthermore, this is reiterated in the recent development in standards (such as ISO 10218), shifting the understanding of what is "collaborative robotics". The term "collaborative robot" now in fact has been phased out , and replaced with "collaborative application". Hence the collaboration is defined by the design of the overall application system and how humans and robots interact within it. This conceptual transition in perspectives underscores that it is the system-level integration, including workspace layout, task design, safety features, and interaction modalities, that determines how effective and safe human-robot collaboration can be.

Table 2.2: Industrial HRI levels.

Level	**Interaction**	**Description**
L0	Fully Programmed	Traditional approach with physically restricting cages; no consideration of HRC.
L1	Co-existence	Agents are separated by safety zones; the robot pauses operation if a human enters the area.
L2	Assistance	Robot assists the human in a task (e.g., lifting heavy objects) but has no independent goals.
L3	Co-operation	Agents work toward a common goal in a shared intervention zone without sharing the same task or physical contact.
L4	Collaboration	Humans and robots autonomously share the task, workspace, and resources to achieve the same goal.
L5	Fully Autonomous	Robots operate independently using ML-trained manipulators; no human intervention is considered.

2.1.4 Metrics in HRI

Evaluating Human-Robot Collaboration (HRC) or Human-Robot Interaction (HRI) is a complex process, largely due to the absence of a universally accepted evaluation framework. The choice of evaluation method is highly application-dependent, with most benchmarks being task-specific and tailored to highlight distinct aspects of a given interaction. Researchers commonly distinguish between performance-related and human-centered dimensions when evaluating robotic systems or collaborative tasks [19].

Evaluation typically involves a combination of quantitative and qualitative metrics. Quantitative measures often include task completion time, success rate, error frequency, and efficiency of robot or human actions. However, the lack of comprehensive, multi-task benchmarks has led to greater reliance on qualitative and subjective

evaluation methods. As noted by [127], some of the most commonly used instruments include the Godspeed Questionnaire, NASA Task Load Index (NASA-TLX), System Usability Scale (SUS), and Robot Social Attributes Scale (RoSAS). These instruments assess perceived trust, workload, usability, likability, and other social or cognitive attributes of the robot. Collected data is typically analyzed using standard statistical methods, such as analysis of variance (ANOVA), to determine significance across conditions or user groups.

In more advanced settings, especially those involving immersive environments, behavioral and physiological metrics – such as gaze tracking, heart rate variability, and other biosignals – may also be used to assess cognitive load, engagement, or stress levels during interaction. While these are less common, they represent a growing area of interest in human-centered HRI research [104].

2.2 Digital Twins

Digital Twins (DTs) are a key component of modern Industry 4.0 bringing physical and digital assets and processes closer. This section outlines the definition and fundamental concepts of digital twins and cyber-physical systems, as well as their applications.

2.2.1 Definition

Although there is no well-established definition of a Digital Twin (DT), it generally refers to a system that includes three main elements: a real space, a virtual space, and communication between the two [67]. The "real space" refers to the physical object or system being modeled, such as a robot, while the "virtual space" is the digital representation of that object, typically consisting of 3D models, sensor data, and simulations. Communication between these two spaces enables real-time synchronization, allowing for continuous updates to the virtual space based on the state of the physical system. DTs have applications extending beyond the manufacturing industry or robotics into other domains such as smart cities, construction, healthcare, and more.

2.2.2 Characteristics and Applications

As described in the section above, the digital twin is defined as a virtual representation that mirrors the structure, context, and behaviors of the individual physical assets or complex systems [29]. The digital twin is continuously updated with real-time data from the physical system. It's crucial to highlight that the DTs go beyond just traditional modeling and simulations, since they provide a unique digital identities to the physical assets and processes [46, 29]; Figure 2.2 shows the conceptual representation of the digital twins. The reviewed literature reveals several key characteristics of digital twins, which are summarized and analyzed in the following list:

- Bidirectional Interaction: The central and vital aspect of the digital twins is the uninterrupted two-way communication and feedback loop between digital and physical entities. The information must flow from the physical system to update the virtual twin, and similarly the information should flow the other way around to inform decisions and trigger changes [29].

- Dynamic updates: The virtual twin must be dynamically updated with data from the physical system in real time [114]. Since the acquired data can be heterogeneous and large in scale, it must be appropriately processed and tailored to ensure accurate, timely, and efficient integration into the digital twin model.

- Mimicry and Representation: A Digital Twin is expected to entirely replicate the structure, behavior, and state of its physical counterpart across spatial and temporal dimensions [46]. This representation is not static – it continuously reflects the evolving conditions of the physical system, capturing both observable and latent variables. High-fidelity mimicry allows the virtual twin to mirror complex dynamics and enables predictive insights. Depending on the domain, this may include geometry, material properties, operating conditions, or interaction with other systems.

- Informs Decisions: A digital twin ultimately informs decisions that realize value. They support decision-making processes and can enable autonomous decision-making or support human-driven judgments [114, 29].

- System-of-Systems: A Digital Twin is described as a system-of-systems that goes far beyond traditional computer-based simulations and analysis. It is a replication of all the entities, processes, and firmware of a physical system into

a digital counterpart [67]. The definition of a digital twin, as proposed by the NASEM report, refers to mimicking "the structure, context, and behavior of a natural, engineered, or social system" – i.e. the system-of-systems. This phrasing is used to describe digital twins of physical systems in the broadest sense possible, encompassing the engineered world, natural phenomena, biological entities, and social systems [114].

- Integration of Models and Data: Digital Twins are different from the traditional computer-based simulation by how the models and data interact. In a DT, virtual models – ranging from physics-based to data-driven machine learning approaches – are blended with real-time data from the physical system. This continuous synchronization ensures that the digital representation evolves alongside its physical counterpart. The choice between model-centric, data-centric, or hybrid approaches is guided by data availability and the nature of the application [29, 67].

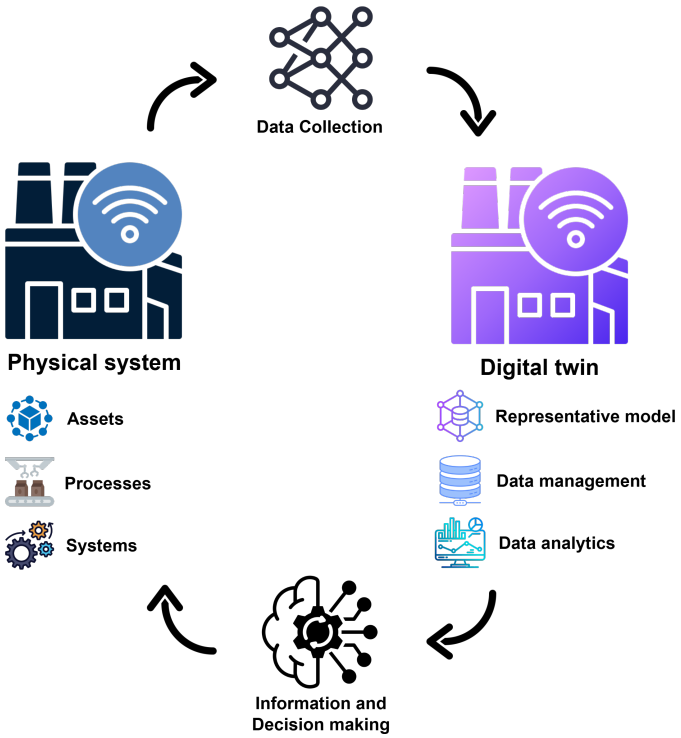


Figure 2.2: Conceptual Representation of a Digital Twin

Digital Twin is considered one of the most important pillars of Industry 4.0, and As mentioned in section 2.2.1, the potential application of digital twins are vast across various domains. In mechanical and aerospace engineering, they accelerate development, reduce risk, predict system failures, and lower sustainment costs – especially in applications like aircraft and spacecraft structural health monitoring [40, 29]. In manufacturing, digital twins enhance first-time yield, enable design-for-manufacturing optimizations, and streamline factory operations [114]. They support both product-related twins, which automate manufacturing and inspection processes, and factory optimization twins for real-time logistics and lifecycle management [46]. During operations, digital twins improve availability, reduce maintenance costs, and support predictive maintenance and anomaly detection through real-time monitoring. In human-robot collaboration, digital twins simulate interactions to enhance system safety and resilience [120]. In healthcare, patient-specific twins aid in optimizing treatments, such as personalized radiotherapy [40]. In the built environment, they monitor infrastructure like bridges and assess sustainability in structures such as railway stations. Their utility extends to energy, utilities, additive manufacturing, agriculture, and even smart cities, providing robust tools for monitoring, fault diagnosis, and decision-making under uncertainty [67, 46]. AR/VR technologies augment digital twin interfaces, enabling immersive interaction and monitoring [48, 41]. Finally, at the end of an asset's life, the digital twin can be archived for historical analysis or reused if the asset is repurposed, preserving critical lifecycle data [29].

Digital Twins in Robotics

The integration of digital twins in robotic systems enables advanced functionalities. DTs serve as simulation environments where researchers can analyze and optimize real-world processes in a controlled and risk-free setting. This capability is particularly useful since testing physical prototypes can be costly or dangerous. Additionally, DTs have been explored as tools for manipulator validation, enabling accurate modeling and performance evaluation for industrial robots [56]. DTs can also be used as virtual representations for monitoring robot health, predicting failure, and optimizing maintenance schedules [108]. When combined with machine learning, DTs serve as the necessary infrastructure for processing vast amounts of data and enabling AI-driven decision-making [41]. Through predictive analytics, they support applications like fault detection, optimization, and intelligent decision-making.

The integration of digital twins with immersive technologies (XR) further amplifies their potential (see Section 2.3 for more on XR). It enhances human-robot interaction by offering intuitive interfaces that bridge physical and digital representations, enabling users to monitor, simulate, and control robotic systems in an immersive and interactive manner [67, 10]. The convergence of digital twins, robotics, and XR opens up new possibilities for creating responsive, adaptive, and immersive control systems [120]. However, challenges remain in areas such as data synchronization, real-time performance, and interoperability across diverse robotic platforms, which must be addressed for digital twins to fully realize their potential in robotics.

2.2.3 Digital Twins and Cyber-Physical Systems

Both Digital Twins (DT) and Cyber-Physical Systems (CPS) are part of the change driven by the Industry 4.0. Although, the concepts are very similar, they are different in their scope and functionality. Cyber-Physical Systems originate from the embedded systems, emphasizing the close coordination between the physical processes and computation, integrating computation, communication, and control (3C) technologies [4]. CPS is often seen as a system of interconnected DTs, positioning the DT as an enabling technology for CPS [10]. In this view, the DTs provide the intelligent virtual representation and the two-way feedback loop necessary for the sophisticated monitoring, control, and managing capabilities described in the original vision of CPS. The predictive and decision-making capabilities inherent in DTs enhance the monitoring and control functions of a CPS, allowing for things like predictive maintenance, anomaly detection, and optimized operations [105]. The ability of DTs to model and simulate interactions, for example in the area of Human-Robot Collaboration (HRC), directly supports the goals of building resilient and efficient cyber-physical production systems.

While overlapping, a distinction can be seen in their primary focus. CPS broadly addresses the integration of cyber and physical elements for control and monitoring across potentially wide-ranging systems. DTs, while leveraging this integration, are more specifically centered on creating a unique, dynamic, predictive virtual replica of a particular physical entity to inform decisions. A DT is characterized by its mimicry and predictive capability tailored to its specific physical counterpart [67].

The boundary between the two concepts is often blurred due to their overlapping features and shared goal of integrating physical and virtual spaces, the terms are

sometimes used interchangeably. Both involve a close coupling and dynamic interaction between the physical and digital domains.

2.3 Extended Reality in HRC

2.3.1 Immersive Technologies: Core Concepts

The term extended reality (XR) serves as an umbrella concept encompassing the full spectrum of immersive technologies, including augmented reality (AR), virtual reality (VR), and mixed reality (MR). Virtual reality (VR) constructs a purely simulated environment in which users can interact with and perceive virtual objects through an immersive experience [22]. In contrast, augmented reality (AR) overlays digital information onto the physical world, enhancing real-world objects with visual augmentations [103]. Mixed reality (MR) blends physical and digital elements more deeply, enabling dynamic interaction between real-world and virtual components. Some experts view MR as a more advanced form of AR [101]. Overall, these technologies fall along what is known as the reality-virtuality continuum – a concept introduced by Milgram et al. [69] – one of the foundational works in this domain, which captures the varying degrees of immersion from the physical to the fully virtual world (Fig. 2.3). The varying degrees of immersion enabled by XR technologies provide a spectrum of human-technology interaction interfaces, tailored to both digital and physical environments. This flexibility is particularly advantageous in human-robot collaboration and interaction scenarios, where seamless interaction between digital and physical elements is often essential across multiple levels.

Figure 2.3: Reality-Virtuality Continuum, adopted from [69]

Technology Landscape

The current extended reality (XR) ecosystem is a heterogeneous range of devices and development platforms that facilitate immersive and interactive experiences. At the center of XR applications are Head-Mounted Displays (HMDs), which can be further categorized into tethered devices (e.g., Oculus Rift and HTC Vive) that need external computing power, and stand-alone devices (e.g., Meta Quest and HoloLens) that are self-contained. Mobile augmented reality (AR) experiences are also broadly supported via smartphones and tablets utilizing onboard cameras and sensors. Most of the HMDs contain integrated sensors for user gaze tracking or head positioning, cameras, and microphones, hence potentially enabling a multi-modal input within a single device. Aside from HMDs, peripheral hardware such as handheld controllers and haptic gloves are used to enhance user embodiment and interaction in virtual environments by enabling tactile feedback and gesture tracking.

On the software side, XR development is centered around game engines such as Unity and Unreal Engine that offer real-time 3D rendering and physics capabilities. These engines are supported by software development kits (SDKs) for extended reality, such as ARCore (by Google), ARKit (by Apple), Vuforia, and the Meta SDK, which offer vital tools for tracking, environmental mapping, and user interaction. In addition, OpenXR SDK, an open standard created by the Khronos Group, plays a crucial role in providing a cross-platform compatibility, thus allowing developers to build XR applications that run on a variety of hardware and runtime environments.

2.3.2 Real-World Industrial Applications of XR in HRC

This section aims to review recent studies related to XR application areas in HRC and elborate on practical implementations. The reviewed use cases are conditionally divided into four categories for a more organized and comprehensive representation, offering diverse viewpoints through the lenses of operator support and communication, safety considerations, teleoperation, and robot programming.

Operator Support and Communication

The idea of using extended reality for the purpose of operator support is not new, and it is not akin to purely HRC. Virtual and augmented reality have been applied in the areas of product development [7] or operator task training [31]. The authors

in [77] define the uses of XR for operator support in the following ways:

a) Show visual and text information regarding the process.

b) Provide the operator with visual and audio cues warning about certain dangers, such as the movement of the robot

c) Visualize the area used by the robot within the real environment to minimize the risk of collisions.

Overall, the reviewed literature supports the classification by [77]. Bolano et al. [8] presented an interface that visualizes the swept volume of the robot's planned motion using a point cloud, which allowed the operator to foresee the volume that the robot will occupy. Chu et al. in [15] proposed two AR-based visual interfaces to provide human operators with situational awareness. One of the interfaces displayed a semi-transparent barrier next to the manipulator, warning the operator of the robot's working envelope. Another interface displayed the virtual gripper model that moved along the robot's trajectory a few seconds before the physical robot, giving the operator enough time to assess the situation. The study by Dimitropoulos et al. [23] proposed a human-to-robot collaboration interface involving both AR technology and machine learning. The authors deploy a convolutional neural network to an AR headset to assist the operator in detecting the assembly parts of interest. Moreover, the authors position several markers throughout the testing environment, which are then detected by the AR headset, enabling the locating and tracking of operators in the scene, therefore eliminating the necessity for multiple stationary RGB cameras. This approach enhanced the flexibility of operator tracking and provided input data to the manipulator regarding the operator's actions and movements. Furthermore, for the collaborative tasks, the implemented interface included gesture-based commands letting the user modify the end-effector position if required.

It is important to highlight that the classification outlined in [77] primarily emphasized a passive approach, focusing on supporting functions and certain safety aspects for humans in proximity to the robot. Nonetheless, a more proactive perspective would involve considering human-to-robot communication, enabling the manipulator to find the optimal way to assist the human operator. In general, the utilization of extended reality head-mounted displays offers various input modes for controlling the manipulator, including speech, gaze, and hand gestures. Authors in various studies explored these modes individually, as well as the possibility of fusion. In [8],

the authors showed two separate communication interfaces allowing the operator to either use the voice command or point with the hand. Meanwhile, the authors in [12] attempted to let the operators use both voice and hand gestures simultaneously. The proposed framework included the calculation of the confidence score for each communication mode to address controversial input data. Another methodology was investigated by Mukherjee et al. in [72], where authors proposed the AI-powered multi-modal fusion architecture based on fuzzy inference and Dempster-Shafer theory to deal with incomplete or conflicting evidence. The experiments were conducted using voice commands and hand gestures, however, authors claim the model should be sufficiently generalizable to also include other modes of input.

Safety Considerations

In human-robot collaboration scenarios, the operator's safety is a primary concern. Multiple frameworks have been developed based on monitoring separation, speed, power, and force limitations. Lately this study has been further extended by trying to predict possible collisions through the optimization-based control methods. The implementation of extended reality cannot directly solve the optimization-related problems of collision avoidance, but it may provide a flexible solution to increase the operators' safety.

Cogurcu et al. [17] suggested an AR-based virtual safety zone system around the manipulator comparable with cell cages for industrial robots. The virtual barriers are dynamic, changing position relative to the manipulator movements. If a human enters the safety zone, the robot stops immediately. A similar but inverse approach has been taken by Hoang et al. [37], guaranteeing an effective way to track human motion by creating a virtual barrier around the user anchored to the AR headset, allowing the person to move around the workplace freely. In case the robot detects an edge of the barrier on its planned path, it must adapt to avoid collision. The work goes even further and showcases the possibility of adding obstacle-oriented virtual barriers restricting the robot's motion in certain areas. The implementation in [17] and [37] require the operator to wear the XR headset continuously. Potentially, this methodology could be used as a way to gather operator movement data in order to learn and manage the individual operators' preferences (related methodologies are also described in [23] and [14]) or to use in the collision prediction as task-specific historical data.

The authors in [106] utilize AR technology to visualize the robot's working envelope. Furthermore, the virtual twin of the physical robot is visualized to give the user a better idea of the planned manipulator motion and future positions. Meanwhile, [14] proposed a significantly more complex architecture consisting of the robot's digital twin, deep learning model, two depth sensors, and the mixed reality headset. The authors investigate different strategies to extract data on the operator's location to synchronize it with the robot's digital twin. Essentially, the study manages to accurately calculate and visualize the distance between the operator's hands and the manipulator in real time, leading to better surroundings and safety awareness.

Some authors approach the concerns for safety from another corner of the RV continuum – Virtual Reality. Creating a completely virtual environment allows the mimicking of realistic as well as potential hypothetical scenarios [3], where users can interact and familiarize themselves with the equipment at no risk of injury. Additionally, the concept of HRI in virtual reality can be elevated by incorporating the digital twin of the manipulator. Hence, it is no longer just a training simulation, but a real-time teleoperation (discussed in detail in a subsequent section) that blurs the boundary between the virtual and physical interaction [56]. On the other hand, the aspects of mental and physical load of interactions in VR are not fully examined, and concerns of cybersickness should be addressed through further research [100].

Teleoperation

Described as the remote, real-time control of the robot, teleoperation is a widely studied area of research in robotics [99]. The teleoperation process is usually associated with multiple challenges [122]. The first issue is mapping a rather large number of joints and degrees of freedom to the human's control interface. Secondly, poor perception leads to lower situational awareness, therefore influencing the overall ability and efficiency of the operator to accomplish the task. Finally, the task planning process for an operator from a remote location is particularly hard due to the need to breakdown the high-level objective into a low-level sequence of actions. Therefore, this section provides an overview of studies that address the above-mentioned issues by leveraging XR technology.

Kennel-Maushart et al. [50] presented an MR interface for multi-robot systems that allows the operator to specify target poses, avoiding unfavorable setups that lead to singularities. The authors present their optimization method tested on a

dual-arm ABB YuMi via the developed MR interface, allowing the user to teleoperate the payload in real-time and remotely. In order to adjust the orientation, position, velocity, and force of the robot, Sun et al. [102] introduced an MR-based teleoperation interface with an integrated series of fuzzy-based algorithms, improving the overall maneuverability of the system.

One of the most interesting sub-domains for research is multi-view teleoperation, where the operator has access to several points of view, solving problems of occlusions and leading to better spatial awareness. One of the most common and straightforward approaches is picture-in-picture (PIP), where multiple video streams are overlaid simultaneously. Usually, the global view is represented as the third-person view of the system, while the local view is extracted from the camera attached to the end-effector. The primary issue with the PIP method is the need for operators to frequently switch between views, resulting in a continuous change of operating perspectives. A multi-view fusion method is presented in [112] showcasing the possibility to construct a 3D point cloud reconstruction of the objects that are occluded in one of the views. The authors use a VR headset as the basis for their interface. The global view, captured by a stationary stereo camera, is displayed alongside the visual augmentations for the occluded objects (extracted from the local view). Furthermore, the occluded robot components, such as gripper fingers reaching for an object in the box, are also rendered as a visual augmentation.

Kuts et al. in [56] investigated the viability of the digital twin (DT) as the validation tool for industrial robot manipulation. The implemented framework includes the DT of the manipulator in the virtual environment, which is fully synchronized with the physical robot. The VR interface for robot control includes the possibility of changing the joint rotation angles, speed, and gripper function. This approach lets the end users remain in the decision loop remotely and in real time. Additionally, the presented interface in [56] requires the operator to manually modify the position of each joint until reaching the destination, which, in fact, leads to the idea of the task-level authoring [93] – forcing the human to break the objective into smaller steps.

Meanwhile, DelPreto et al. [21] presented an online learning framework where human demonstrations are conducted in order to complement ML-based autonomous robots. The robot uses self-supervised learning, however, if the task cannot be properly accomplished, it request a direct demonstration from a human operator that is performed via Virtual Reality. The work in [21] is a great example of autonomous robotics with the human-in-the-loop, where XR acts as a human-robot communica-

tion middle-ware complementing AI algorithms with the human experience.

Robot Programming

Robot programming, including operations such as relocation, grasping, and orientation change, are all among the most important functionalities of a robot [60]. In general, robot control methods can be divided into traditional and learning-based methods. The traditional control, delivered through offline programming, allows robot actions to be fully programmed. Nonetheless, it lacks the flexibility required in rapidly changing environments where it is nearly impossible to foresee all circumstances. Kinesthetic teaching partially addresses these concerns by enabling the user to easily and directly modify or build from scratch the robot's waypoints, grasping positions, etc. On the other hand, learning-based methods generally employ the use of machine learning algorithms. The implementation of AI opens opportunities for a larger degree of autonomy, better generalization in tasks, and even behavior modeling.

Back in 2012, Fang et al. [26] introduced an interactive framework based on Augmented Reality (AR) for adjusting a robot's path. The authors incorporated their framework with the robot's task and trajectory planner, enabling the operator to review the initial path. This integration offered the flexibility to modify, add, or delete waypoints between the starting and destination points. Quintero et al. [88] presented a trajectory modification interface similar to the one in [26]. However, the authors also conducted a study to compare it to kinesthetic teaching. The findings indicate that AR-based trajectory modification frameworks can reduce the teaching time, and show better overall performance since it is easy to use and is less physically demanding.

Luebbers et al. [63] proposed a method of constrained learning from demonstration with the purpose of long-term skill maintenance of the manipulator. The introduced AR interface allows users to visualize and modify the task-associated gripper positions as well as the movement constraints. Interestingly, the idea of managing virtual constraints (also referred to as barriers) is similar in nature between [63], where authors use it for robotics path planning to accomplish a task, and [37], where the primary subject of interest is safety aspects.

The term HRC often implies an arm manipulator, however, it can also refer to other robotic platforms as well. In fact, many researchers attempt to study the interaction methods with mobile robots, including the role of XR in the process. For

example, Tsamis et al. [106] presented an AR-based framework of a manipulator on a mobile platform. The mobile robot navigates to the goal pickup position, where the arm utilizes object detection to plan its path for grasping. AR plays a key role in keeping humans in the decision-making process by reviewing the planned routes of both agents. Focusing fully on mobile robotics, Gu et al. [34] presented a simple yet effective AR-based interface for navigation goals. The AR Point&Click interface allows the use of natural pointing gestures, which are captured and interpreted by the cameras on the AR headset. The authors compare their approach to several other methods and conclude that based on user study, the proposed interface leads to higher efficiency and reduced mental load. Although the presented implementation in [34] was done for a mobile robot, it easily translates to the arm manipulator setting - a similar example is illustrated in [118] as part of a larger research on MR interfaces for HRC.

As mentioned earlier in this subsection, the use of ML has become widespread in the robotics industry, and in this context, XR also establishes its relevance, particularly within the domain of imitation learning. One of the most famous works in this area was published in 2018 by Zhang et al. [123], showcasing a method to directly map pixels to actions from the demonstrations obtained in the virtual environment. Interestingly, an inexpensive system with less than 30 minutes of demonstration was sufficient to achieve nearly 90% success rate. Similarly, Dyrstard et al. [25] investigated the possibility of skill transfer for fish grasping tasks. By collecting just a few dozen demonstrations in virtual reality and employing domain randomization, a substantial synthetic training dataset comprising 100,000 samples was generated. Considering the given task and setting, the authors managed to achieve 74% accuracy in grasping. After a more thorough analysis and dismissal of non-ML-related failures, the success rate could be estimated at 80%.

2.3.3 Current Limitations and Emerging Trends

Based on the conducted background research, several preliminary insights emerge, highlighting both the advantages and current limitations of extended reality (XR) technologies in the context of human-robot collaboration applications.

Mitigating Risks

The visualization of motion intention and object manipulation gives the operator a better understanding of the workflow. In the case of VR systems, one of the main advantages is the elimination of the need for physical expert presence [2]. In general, studies show that XR is a unique tool that allows the conduct of teleoperation, robot programming, and various operator-supporting functions in a safe and controlled manner. Although XR offers the possibility of finding a balance between operator safety and robot efficiency, the impact on physical and mental health from working with head-mounted displays in HRI tasks necessitates further studies. As mentioned in [100], there are currently no optimal solutions to address all possible side effects like muscle fatigue, motion sickness, and mental overload. Similarly, for human-robot collaborative tasks, the psychological factor remains a significant area of research. This includes the effort to cultivate trust between the agents, as well as methods to generate motion and trajectories that closely mimic human behavior [38].

Immersiveness

One of the bases for introducing XR is its immersive potential. The immersive environment allows users to perceive the spatial aspects of a robot and its surroundings more effectively. By providing a realistic and engaging experience, users can interact with and understand the robot's movements and actions in a way that is not possible through traditional interfaces. This level of interaction and understanding can facilitate better demonstration quality, which is one of the most decisive factors in the effectiveness of robot policy learning [60, 44].

Given the enhanced visualization and situational awareness provided by immersive capabilities, task coordination is another area that can be improved for better collaboration between agents. The application of XR in the context of multi-robot systems can provide easy-to-use, intuitive methods for commanding multiple agents synchronously [51].

User-oriented Concepts

In order to best address human-robot collaboration, it is essential to take social cues into account. That can be expressed in terms of communication modes like gestures, voice, or gaze – all of which are supported by the modern XR headsets. The use of head-mounted displays (HMDs) eliminates the need for multiple stationary cameras

and various additional sensors. However, further investigation is needed on how to capture and accurately interpret multi-modal communication signals, such as fusion techniques. Similarly, the tracking functionality in HMDs opens certain possibilities to study operator preferences. The authors in [74] propose a method to transfer operator preferences from a canonical to an actual assembly task, allowing the cobot to assist the operator proactively. The incorporation of XR in this process could facilitate data collection – operator and surrounding related, and potentially result in a more personalized HRI experience.

From Lab to Industry

One of the primary obstacles preventing the adaptation of extended reality for the human-robot collaboration scenarios is setup costs, requiring substantial initial investment. Additionally, there is a lack of unified consensus within industry on the XR integration strategy and interface development, which understandably stems from the differences in industry-specific requirements and individual products. A possible future area of research could involve investigating the feasibility of utilizing the same XR interface across multiple manipulators with the flexibility to easily customize the interface for new robot- or product-specific characteristics. Furthermore, the implementation of XR is usually performed in a bundle with other digital technologies. Therefore, it entails additional investments and resources, and in fact, the effectiveness of XR becomes conditional on the development and maintenance of other technologies. Also, it is noteworthy that most of the reviewed works perform experiments in the laboratory. The prospect of transferring the developed methodologies and interfaces to real-world scenarios and industrial settings remains yet to be explored.

Chapter 3

Containerization for Digital Twins

The effectiveness of industrial and robotic systems depends on adaptability and scalability, enabling robust system integration and reproducible performance across varying operational conditions. Cyber-physical systems by nature are complex [81], including different hardware components and software applications. Their dependencies undoubtedly undergo periodic upgrading with their associated technologies causing compatibility issues with legacy applications (e.g. version mismatches, deprecated packages and dependency conflicts). One widely used middleware in robotics is the Robot Operating System (ROS) [87], an open-source middleware running on Linux to develop and facilitate communication between different robotic components. Sensor drivers may face compatibility issues even within the same ROS version, and ROS itself is tightly coupled to specific Ubuntu/Linux distributions, further complicating system maintenance. In such scenarios, ensuring a consistent ROS version or Linux distribution on a single host machine becomes increasingly challenging.

To address these challenges, containerization technologies such as Docker, provide isolated and modular environments, allowing developers to manage heterogeneous setups efficiently [6]. This supports the development of scalable and reproducible software pipelines for robotics applications while ensuring cross-platform compatibility [66]. Additionally, containerization reduces system overhead compared to virtual machines, supports networked communication between distributed components, and facilitates version control, allowing developers to quickly test, deploy, and roll back software changes when necessary [96]. These advantages make containers an effective tool for managing digital twins, facilitating synchronized real-time updates and remote deployment across diverse hardware platforms.

3.1 Background

3.1.1 Robot Operating System

The Robot Operating System is an open-source framework designed to simplify robotic software development by providing various standardized tools and libraries [87]. An important feature of ROS is its messaging-based communication, which follows a publisher-subscriber model to facilitate interaction between modular system components. These components are symbolized by nodes, which are independent processes responsible for specific tasks such as sensor data processing, motion planning, or actuator control. By structuring a robotic system as a collection of modular nodes, ROS allows for greater flexibility, scalability, and ease of maintenance.

The basic building blocks for ROS communication include: topics (publish-subscribe messaging for continuous data streams), services (request-response for on-demand interactions), and actions (goal-oriented tasks that provide feedback over time). Integrated within ROS are several toolkits to simplify common robotics tasks. MoveIt is one of the most widely used toolkits that enables path planning, collision avoidance, and inverse kinematics (IK) solvers for robotic manipulators [18]. These features, along with many others, make ROS a powerful and flexible platform for creating and managing robotic systems, digital twins, and immersive control applications.

3.1.2 Virtual Machines vs Docker

Virtual Machines (VMs) have been widely used to overcome challenges in developing complex systems by replicating the functionality of physical hardware [97]. They function as standalone computers running on a host machine, with their own operating system (OS), applications and virtualized hardware, all managed by a hypervisor. However, VMs can be resource-intensive, slower to start, and may require significant computational power when running multiple instances [84]. For example, a system might require two VMs with different OS and ROS versions to accommodate varying dependencies. This can degrade system performance, especially for real-time robotic control or simulation tasks that demand deterministic execution.

These limitations led to the exploration of alternative solutions, such as containerization technologies, to build more robust, efficient, and flexible systems for robotics. Docker offers a lightweight solution by packaging applications along with all their dependencies into containers [113]. Unlike VMs, containers leverage the host OS ker-

nel, eliminating the need for a hypervisor, which makes them lighter, faster to start, and less resource-intensive. Docker containers are created from Docker images, which serve as blueprints of the virtualized environment, specifying all necessary configurations, dependencies, and system resources. Dockerfiles[1] define these images, providing a streamlined and standardized way to build and manage environments which can be replicated quickly. This enables faster deployment cycles compared to VMs, which are slower and require more resources to configure.

Furthermore, Docker's lightweight nature allows for the rapid launch of multiple containers from different images, significantly faster than starting multiple VMs. Tools like Docker Compose make managing complex architectures easy by using a single YAML configuration file to control services, networks, and volumes. This improves developer agility and enhances system management. Docker is also free to use and automatically stores images locally on the host device, further improving its accessibility. It offers developers enhanced agility due to its speed and simplicity. One of its primary advantages is portability: by sharing the Dockerfile and Docker Compose file within teams, developers can ensure that environments are consistent across different machines, regardless of hardware specifications. This reduces system-specific issues and fosters collaboration.

3.2 Implementation

As part of this thesis, a cyber-physical system was developed, centered around a digital twin of the research lab featuring a collaborative robot – the Kinova Gen 3, a 6-degree-of-freedom robotic arm[2]. This system brings together three key components: the physical robot, its virtual representation in an immersive 3D environment, and XR-enabled human control – all connected via bidirectional communications. Docker is used to create a consistent, modular environment containing all necessary ROS packages, robot drivers, and software dependencies. The robot's digital twin is implemented within the Unity game engine, serving as a real-time, interactive visualization of the robot's state and actions. Figure 3.1 presents a visual overview of the system's network architecture, with each component explained in detail in the sections that follow. Furthermore, the full source code of the implementation is made

[1] https://docs.docker.com/build/building/best-practices/
[2] https://www.kinovarobotics.com/product/gen3-robots

public for the research community as a GitHub repository[3]. More detailed information about the Unity engine, its integration with XR, and the user interactions is provided in Chapter 4.

The Figure 3.1 illustrates the full communications pipeline: On the client side, the Dockerfile specifies how to build the image, while Docker Compose is used to send commands to the Docker daemon to execute the build. If necessary, base layers such as Ubuntu and ROS are pulled From Docker Hub/Registry. Running `docker-compose build` and `docker-compose up` directs the Docker Daemon to create and start the container from the built image. The container then launches ROS nodes to enable communication between the robot's digital twin, ROS, and the physical robot.

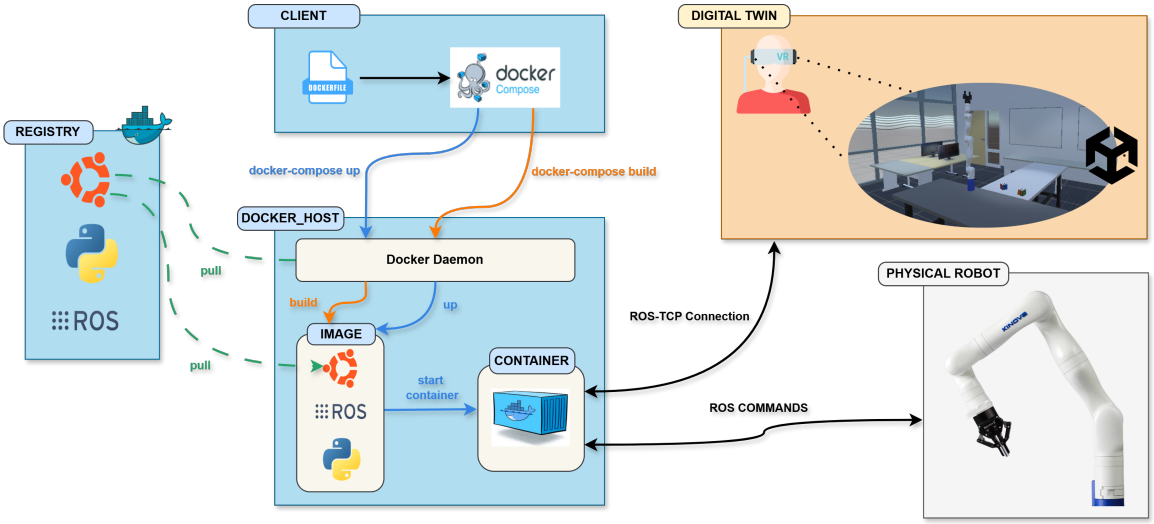


Figure 3.1: Communication Pipeline for the Cyber-Physical System

3.2.1 Containerized Environment

The pipeline development follows the best practices outlined by Melo et al. in [66]. The following principles form the foundation of the system design:

- **Reproducibility**: The system can be replicated across various hardware setups.

[3]https://github.com/ykarpi/Gen3-Immersive-Control

- **Suitability**: The design is adaptable to common computing platforms.
- **Ease of Integration**: The system allows for seamless addition or substitution of devices with minimal modifications.

Docker Setup

The Docker image is built incrementally in a layered way, with each layer defined by specific commands in the Dockerfile. The base layer `ros:noetic-ros-core-focal` is a minimal ROS Noetic installation atop Ubuntu 20.04 (Focal Fossa). The following layers provide essential tools, dependencies, and configurations such as Kinova drivers, Unity ROS-TCP endpoint, MoveIt and multimedia libraries. The final layers build the catkin ROS workspace and source the environment finalizing the system setup. Hence, a container is ready for use as soon as it's created.

Docker Compose is used to preconfigure environment settings, set IP and port addresses for the ROS master, enable display forwarding to run Linux native applications like RViz (Figure 3.2) and Gazebo within the Docker container on Windows machines, and start the container with its corresponding name. From within the Docker container, the robot's drivers, as well as other packages (such as MoveIt or the TCP Endpoint for Unity) can be launched.

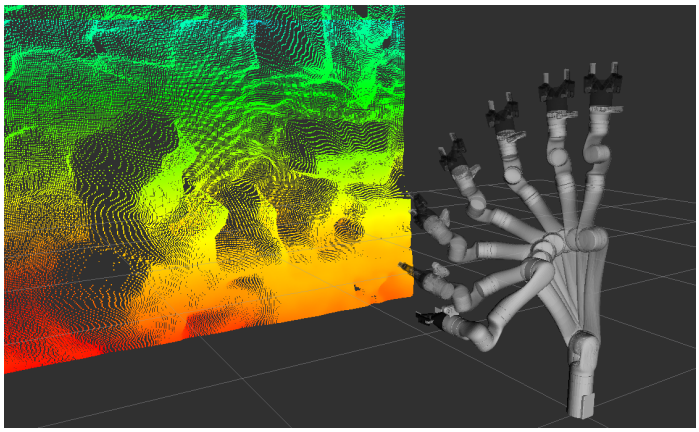


Figure 3.2: Robot Control Using RViz: Motion Planning Is Handled by MoveIt, While the Point Cloud Captured by the Onboard Camera Is Visualized in the Scene

3.2.2 ROS Network

The ROS network is deployed within a Docker container, enabling bidirectional data flow throughout the digital twin. Figure 3.3 illustrates the key nodes and topics, along with the corresponding publish/subscribe communication flow:

- The `/joint_states` topic publishes the robot's current joint position data.
- The `/unity_endpoint` node handles data exchange between the Unity-based digital twin and the physical robot system.
- Commands from the Unity interactive environment to the physical robot are transmitted using two approaches:
  - Directly via Kinova driver topics, such as the `/in/` namespace for emergency stop and fault-clearing commands;
  - Indirectly via the MoveIt framework: the `/unity_to_gen3` node subscribes to Unity-generated command topics (e.g., `/unity`) and forwards them to MoveIt for motion planning and execution.

3.2.3 MoveIt Toolbox

The MoveIt [18] framework is integrated into the pipeline, with the `move_group` node launched alongside the Kinova driver. Additionally, a separate node (`/unity_to_gen3`) receives messages from Unity for controlling the physical robot. The implementation includes several custom Python scripts tailored for the Kinova Gen 3 manipulator. Key functionalities, such as sending the robot to specific positions and moving it to joint configurations derived from the digital twin environment, are implemented using the MoveIt Python interface: `moveit_commander`. Figure 3.3 summarizes the running ROS nodes and topics, illustrating the data flow. By adding further functionality into the system, the network and the flow of data that is shown in Figure 3.3 can be easily modified. Since most motion planning commands are executed using MoveIt, the solution is generalizable and can be readily adapted for other manipulators. Figure 3.4 showcases the high level overview of the MoveIt `move_group` node that acts as an integrator, pulling together all ROS parameters, data channels, and user interfaces.

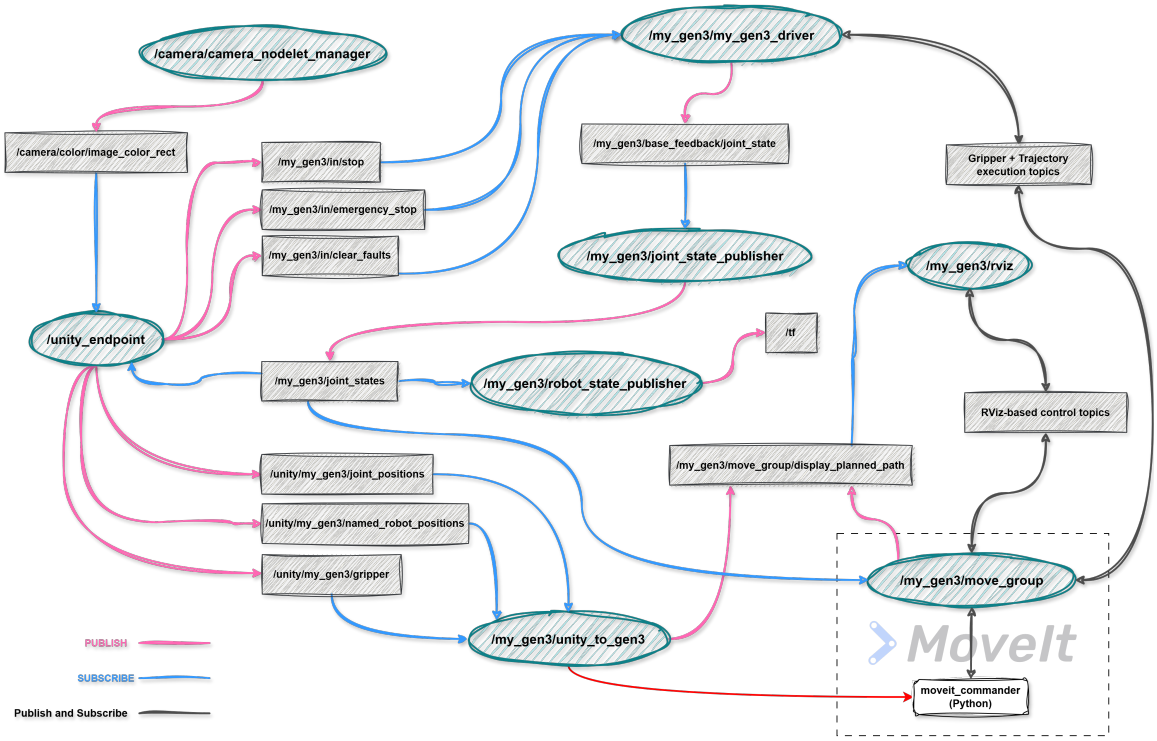


Figure 3.3: ROS Network Running in the Docker Container

3.2.4 Integration with Unity

Unity is a widely used game engine in research, particularly for modeling digital twins [62], due to its built-in physics engine, cross-platform support, and extended reality (XR) capabilities. Its integration with ROS makes it especially suitable for robotics and industrial automation applications. A more detailed description of the Unity engine and its role in XR technologies is provided in Chapter 4.

As shown in Figure 3.1, communication between ROS and Unity is established using the ROS-TCP Connector[4] and ROS-TCP Endpoint[5]. The ROS-TCP Connector is a Unity package that manages communication, message serialization, and request handling on the Unity side. On the Unity side, the IP address and port of the ROS master node must be specified. On the ROS side, the ROS-TCP Endpoint must be launched to handle incoming requests, deserialize messages, and route them to the appropriate nodes. This setup enables real-time, bidirectional communication between Unity and the ROS network, effectively linking the digital twin to the physical robotic system.

[4]https://github.com/Unity-Technologies/ROS-TCP-Connector
[5]https://github.com/Unity-Technologies/ROS-TCP-Endpoint

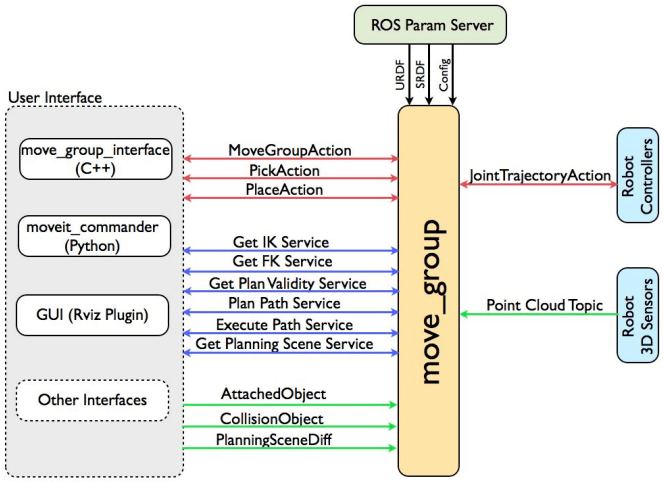


Figure 3.4: The High-Level Architecture of the MoveIt `move_group` [80]

3.2.5 Simulation Tools: RViz and Gazebo

One of Docker's advantages is its ability to run applications like RViz and Gazebo on a Windows machine. Display forwarding in Docker is part of runtime configuration and is not part of the image construction. When an image in Docker is built with a Dockerfile, the contents of the image – its operating system, libraries, and application code – are established but not how the container will communicate with the host system at runtime. On the other hand, display forwarding typically is done during the container startup, utilizing tools such as `docker-compose`. This configuration involves the definition of environment variables like `DISPLAY` so that the host X11 server can be accessed by the container. Such capability is especially needed when executing graphical programs, such as RViz or Gazebo on ROS setups. By setting up display forwarding in the `docker-compose.yml` file, the container can display graphical user interfaces on the host machine.

Figures 3.5 and 3.6 illustrate the visual output of RViz and Gazebo, respectively, when executed within the Dockerized ROS environment. These simulation tools allow for flexible configuration, including the ability to attach different end-effectors to the robot model. As part of the implementation, the Robotiq 2F-85 gripper was used throughout the thesis to accurately reflect the physical system. Figure 3.5 shows the robot model in RViz, where live sensor data such as point clouds and joint states can be visualized, facilitating real-time monitoring and debugging. Figure 3.6 presents the robot within a simulated 3D environment in Gazebo, enabling physics-based testing of motion planning and control algorithms. Gazebo integration is particularly valuable

in this context, as it supports the development of immersive control interfaces (see Chapter 4) and allows for complete simulation of robot mechanics – without requiring a connection to the physical system.

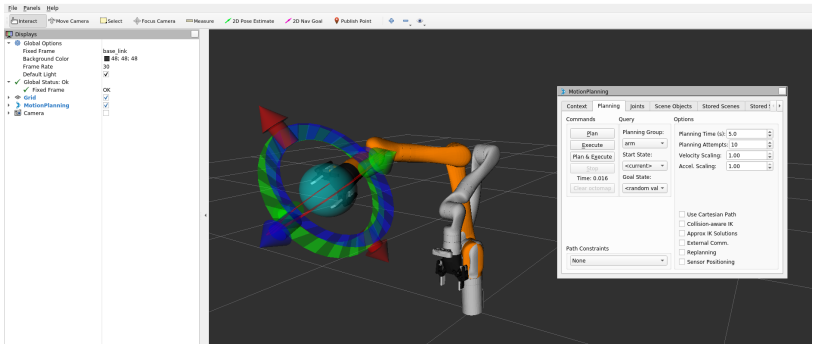


Figure 3.5: RViz-based Control of Kinova Gen 3

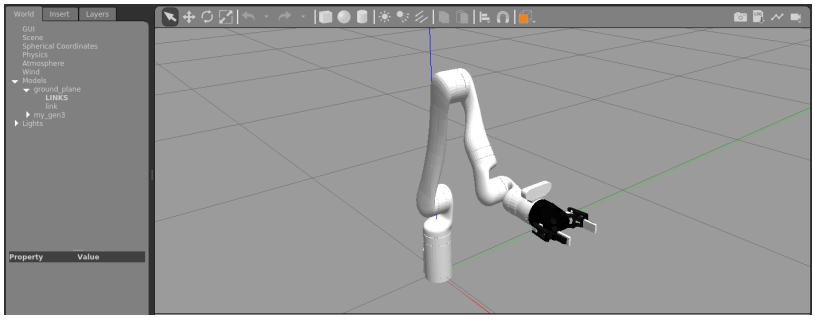


Figure 3.6: Gazebo Simulation of Kinova Gen 3

3.3 Evaluation

For the purpose of evaluating the Dockerized application, Table 3.1 summarizes different categories, their importance, and some of the methods that allow for evaluating a Docker container's performance. The table is divided into several categories: Performance (focused on computer resource usage), Real-Time Performance (focused on data processing delays such as latency), Networking (focused on communication efficiency, including network latency and throughput), Reliability (used for troubleshooting unstable container performance, including restart counts and error logs), Integration (ensures required ROS-related nodes and topics are accessible within the container environment), and Robot-Specific (focused on real-time metrics related to control accuracy and sensor data).

Table 3.1: Evaluation Metrics for a Dockerized Robotic Application

Category	**Metrics**	**Tools**
Performance	CPU, RAM, Disk I/O, GPU usage	`docker stats`, `htop`, `iotop`
Real-Time Performance	Latency, Jitter	`rostopic delay`
Networking	Latency, Packet Loss, Throughput	`ping`, `iperf`, `netstat`
Reliability	Restarts, Data Persistence, Error Logs	`docker logs`
Integration	Hardware access, ROS topic/service health	`ls /dev`, `ros topic list`
Robot-Specific	Control Accuracy, Sensor Data, Real-World Latency	`rviz`, `rostopic echo`

For the presented in this chapter implementation, the metrics for the computer utilization were measured using the `docker stats` command. The data was recorded for 30 seconds, and its summary is presented in the Table 3.2. The evaluation was performed on a local PC (the specifications are given in Table 3.3).

Table 3.2: Resource Usage Summary for Docker Container

Metric	**Value**	**Unit**
Average CPU Usage	185.1	%
Peak CPU Usage	204.4	%
Min CPU Usage	89.96	%
Memory Usage	1.11	GiB
Memory Usage (%)	3.56	%
Total Net I/O	12.8/106	GB (RX/TX)
Block I/O	0/0	B (Read/Write)
Active PIDs	327	-

The following list elaborates on the key metrics from the Table 3.2 and their meaning:

- CPU Usage: The container is consistently using a very high amount of CPU, ranging from approximately 89.96% to 204.40%, with most readings above 170%. This indicates intensive processing activity, such as from real-time robotics or vision computation tasks. CPU usage can exceed 100% if Docker is using multiple CPU cores. For example, 200% = using 2 full cores.
- Memory Usage: The memory usage remains stable at around 1.11 GiB, which is only approximately 3.56% of the available 31.29 GiB. This suggests that the container is not memory-intensive.
- Network I/O: A total of about 12.8 GB received and approximately 106 GB sent. The high outbound traffic suggests that the container is streaming or uploading large volumes of data, for example, from robot sensors or camera feed.
- Block I/O: No block I/O activity was recorded (0 B / 0 B), indicating no disk read/write operations. This is expected, as the application did not mount any volumes to the host machine's file system.
- Process Count (PIDs): The number of active processes within the container remains stable, hovering around 326-327. This relatively high count is expected in ROS-based systems, where multiple nodes and threads may run concurrently.

Another important metric to assess was the system's real-time responsiveness and latency. For instance, the delay of a critical topic, `my_gen3/joint_states` - responsible for the robot's joint positions - was measured and found to be consistently stable at approximately 1 ms. The observed latency value falls well within industry standards. However, during interaction with the robot's digital twin in the virtual environment, a slight delay is still noticeable. This latency is likely attributable to the ROS-TCP connection established between the Unity engine and the ROS network. This topic is in fact explored by other researchers in depth, such as in [1] and compared to other existent methods for bridging ROS and Unity environment. Additionally, further delay may arise from the processing of received data within Unity's C# scripts.

The Dockerized robotic application was developed and evaluated on a local PC, with the corresponding hardware and software environment summarized in Table 3.3.

Table 3.3: Hardware and Software Specifications Used for Container Evaluation

Component	**Specification**
Host OS	Windows 10
Docker Version	27.2.0
Container Base OS	Ubuntu 20.04
ROS Version	ROS Noetic
CPU	AMD Ryzen 9 5950X, 16 cores
RAM	64 GiB DDR4
GPU	NVIDIA Titan RTX with 24 GiB VRAM

3.4 Remarks and Future Development

Future work will focus on extending the current architecture to support a broader range of robotic platforms, including autonomous ground and aerial vehicles. Each robot will be managed within its own dedicated Docker container, which will be responsible for receiving high-level commands from the digital twin and translating them into control instructions specific to the hardware. This containerized approach ensures process isolation, improves fault tolerance, and maintains seamless communication across the system, thereby enhancing the scalability and robustness of the architecture for multi-robot scenarios. Furthermore, the integration of artificial intelligence (AI)-driven decision-making should also be considered, with the objective of enabling more autonomous and context-aware behavior within digital twin environment.

While the current implementation demonstrates a working proof of concept, there are several opportunities for streamlining and optimizing system deployment. In particular, automating the startup sequence of ROS nodes and drivers using an updated `docker-compose` configuration – with clearly defined `entrypoints` – would eliminate the need for manually executing multiple launch commands, thereby improving usability and maintainability. Lastly, a key insight gained from the development process is the importance of adopting ROS2 from the start as the foundation for future implementations. Given that ROS1 is approaching end-of-life, transitioning to ROS2 would offer long-term benefits in terms of performance, scalability, and compatibility with emerging technologies.

Chapter 4

Human-Centered Immersive Control for Robotics

Within the conceptual principles of Industry 5.0, human-robot collaboration (HRC) aims to find solutions enabling humans and robots to work together side-by-side, engaging on both physical and cognitive levels. Additionally, the recent advancements in extended reality (XR) technology, encompassing both hardware and software, as well as its integration in digital twins, shows a promising solution to support human involvement as an active agent for HRC purposes [82] or within the broader context of the smart factory concept. Moreover, the integration of machine learning with extended reality offers promising potential for enabling human operators to intuitively instruct robots [48].

This chapter explores the integration of extended reality (XR) technologies in human-robot interaction. Section 4.1 presents the theoretical foundation of the XR-based human-in-the-loop framework, outlining methodologies and potential applications of XR as an interaction medium that involves the human operator in collaboration with an autonomous manipulator.

The remaining sections 4.2 and 4.3 of the chapter focus on the practical implementation of immersive robot teleoperation techniques in both virtual and mixed reality environments. The presented implementations demonstrate the utility of existing approaches and technologies, while also highlighting the current benefits and limitations of immersive technology in robotics across different levels of the reality-virtuality (RV) continuum.

4.1 Human-in-the-Loop Framework Conceptualization

The present section (4.1) introduces a conceptual framework for XR-enabled human operator involvement in the robot learning process. As originally presented in Chapter 2, Table 2.2 outlines a classification of human-robot interaction levels, adopted from [71]. Building on this classification, an intermediate sublevel between levels four ("Collaboration") and five ("Fully Autonomous") is proposed, termed *fully autonomous with human-in-the-loop*. While this sublevel does not require continuous human-robot interaction, it enables human intervention when necessary, augmenting the autonomous system with human expertise (Figure 4.1). The original industrial HRI levels classification by Mukherjee et al. [71] is provided in Table 2.2). The defining features of the proposed sublevel include:

a) Human involvement in robot learning process.

b) Human-Robot skill transfer.

c) Use of Extended Reality (XR) as a communication interface.

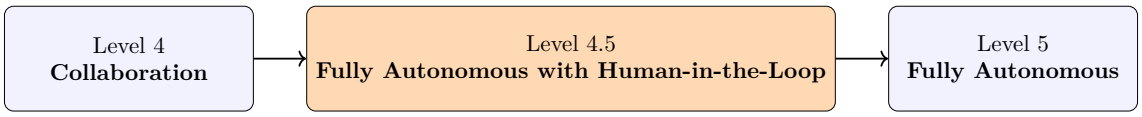


Figure 4.1: Proposed HRI level: *Fully Autonomous with the Human-in-the-Loop*

Defining autonomous robotics is a critical aspect of this work. Hence, Section 4.1.1 is dedicated to introducing the machine learning-based manipulator, and the respective framework for tasks generalization. Simultaneously, the subsequent section (4.1.2) delves into the integration of the human component through XR, specifically focusing on collaboration, programming, and performance oversight. Finally, Section 4.1.3 offers a brief overview of the key technologies whose development could support the implementation and further extension of the proposed framework.

4.1.1 Manipulator Task Generalization

In this subsection, a high-level overview of the ML-based autonomous manipulator is presented, focusing on its autonomy and capacity for generalizable operations. Note

that a detailed discussion of the underlying technologies and methods enabling this system goes beyond the scope of this thesis. The primary focus remains on the human-in-the-loop, XR-enabled component of the system.

It is important to emphasize that the autonomous robot is equipped with pre-trained skill policies. Hence, the Task Generalization Framework (TGF), shown in the center of Fig. 4.2, is designed to increase the manipulator's ability to generalize. The proposed task generalization framework for the manipulator is constructed of a hierarchy of modules and can be largely divided into three major sections: Demonstration, Task Planner, and Task Execution – the subsequent sections briefly elaborate on each of them.

Demonstration

To facilitate the robot's learning of a new task, a demonstration, also known as a sample task, is provided. This demonstration serves as a representation of the presumed task that the manipulator is expected to perform. The demo task, along with essential information about the actual task or the manipulated object, is provided to the next module, the Task Planner. Depending on the real scenario, the essential information may include numerical measurements, colours, shapes, CAD models, etc.

Task Planner

It is assumed that the actual task executes similar skills compared to the demo task, but in a different environment (e.g., varied toolsets, obstacles, and parametric characteristics). Therefore, the Task Planner decomposes the provided demonstration (i.e., sample task) into multiple skills, which are then associated with and utilized for the execution of the actual task. Usually, skills are defined as low-level abstractions or primitives [92]; for example, the task is decomposed into the commands to "move linearly", "locate the object", "position the gripper", "pick up the object", etc. This decomposition forms the foundation of the skills library, a collection of fundamental actions essential for task completion. By providing crucial information, the skills library enhances the efficiency of task execution in different environments.

Task Execution

In the Task Execution module, the agent builds up a heterogeneous spatial representation to localize itself in the environment. Using the spatial representation, the

information received from the Task Planner, and the pre-trained skill-specific policy, the trajectory planner maps the robot motion and carries out the instructed skill.

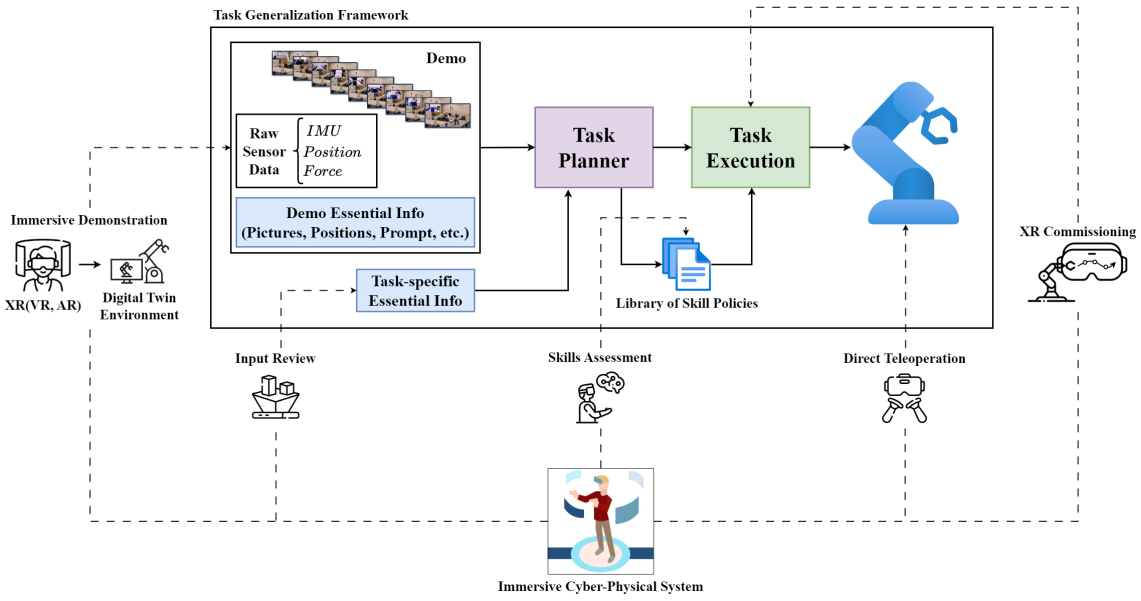


Figure 4.2: XR-Enabled Human-in-the-Loop Approach for Enhanced HRC

4.1.2 Human-in-the-Loop Component

The use of immersive technologies that augment or fully replace the real world creates new opportunities for safe and intuitive human-robot interaction, as well as for task programming. The XR-based interaction approaches discussed here are designed to integrate human intelligence into the robot's learning process, with the aim of supporting task generalization. Moreover, the methods described below are readily transferable to a variety of pre-trained autonomous manipulators. The conceptual foundation for each method outlined in Figure 4.2 is further elaborated.

Immersive Demonstration

In the proposed concept, human experts deliver their skills through immersive demonstrations: an approach that offers a convenient, intuitive, and safe way to illustrate a given task. These demonstrations can be performed entirely within a virtual environment, where all relevant variables are controlled. Immersive demonstrations in virtual reality may serve as the sole source of training data or be used to complement prior

demonstrations, whether conducted in physical or virtual settings, to improve the robot's ability to generalize and perform a broader range of tasks. Furthermore, virtual demonstrations eliminate constraints related to the operator's physical location, enabling flexible and remote task programming and robot teaching.

Another extension of this method is the on-site AR-based programming. The demonstration using AR, conducted on the manipulator's virtual model, could be interpreted as a virtual kinesthetic teaching [43]. It does not have to interrupt the production process since the operator's manipulations are applied to the virtual twin of the physical system.

Input Review

The opportunity to review the task-specific information ensures the accuracy and relevance of the given data, refining the overall process for optimal performance and adaptability in task execution. The exact implementation varies on the type of fed data; it may be visualized through AR or within VR space for the operator to review, manipulate, and modify.

Skill Assessment

The breakdown and abstraction of skills can be revised based on real-time insights overseen by the human. The option to visualize and review the work done by the Task Planner module gives some level of transparency to the machine learning black box nature. The human operator can modify existing skills or incorporate new ones using XR as a virtual programming interface.

Direct Teleoperation

The opportunity to assume complete control of the manipulator through a VR headset enables humans to directly coordinate the robot's actions for task completion. This capability may be particularly useful in cases where the manipulator repeatedly fails to accomplish the task autonomously. Additionally, the requirement for the operator to be physically present on site is eliminated, as teleoperation can be carried out remotely. Furthermore, the experience gained through direct teleoperation could be leveraged to adjust or refine the manipulator's skill policies – for example, through the use of semi-supervised learning methods that incorporate both labeled demonstrations and autonomous behavior.

XR Commissioning

The ability to program the robot using Mixed Reality serves as an effective validation tool, allowing the human operator to review and adjust visualized paths and trajectory parameters. This approach, referred to as XR commissioning, is particularly valuable during the initial deployment of a new manipulator or when modifying existing robot programs. Once a modification is applied, it effectively establishes a new ground truth for retraining the robot's trajectory planner – embedding human expertise into the learning loop and refining the planner's future decision-making.

4.1.3 Enabling Technologies

This subsection aims to give a broader context for how the advancements of particular technologies could influence the future of human-in-the-loop frameworks. While Extended Reality (XR) serves as the primary interface within the presented system (addressed in detail in Fig. 4.2), the focus here is on complementary technologies that support, extend, or enhance XR-driven interaction and control.

Digital Twin

The digital twin concept involves creating a dynamic digital representation of a physical system, enabling simulation, analysis, and optimization. In HRC, digital twins can be used to design and test collaborative processes, predict maintenance needs, and improve system adaptability [28]. Furthermore, the digital twin is a key concept of cyber-physical systems that provides real-time control and monitoring, essential for certain aspects of XR.

Artificial Intelligence

Artificial Intelligence, leveraging machine learning algorithms and models, enhances task planning, decision-making, and environmental perception within HRC. This enables robots to intelligently interpret human gestures, speech, and intent while adapting to their surroundings for more strategic planning [71]. Such AI-driven capabilities ensure seamless adaptation to human behaviors, deepening interaction and improving cooperation, making HRC systems capable of executing sophisticated, context-aware actions [45].

Cloud Computing

Cloud computing provides the infrastructure for scalable and on-demand computing resources, critical in managing the extensive data generated in industrial HRC settings. It enables the centralization of data analysis and storage, offering robust platforms for AI and machine learning models to operate efficiently. This technology strengthens the flexibility of HRC systems, enabling them to adapt to new tasks and environments quickly by leveraging cloud-based data and computational power [42].

Edge Computing

Edge computing processes data near its source, reducing latency and enabling real-time responses critical for human-robot collaboration. By decentralizing computation, it ensures swift data analysis, essential for tasks needing immediate feedback. This enhances robotic autonomy, safety, and operational efficiency, especially in environments where split-second decisions are crucial [95]. Edge computing's integration into HRC systems supports seamless operation and higher responsiveness, aligning with the demands of advanced manufacturing and collaborative tasks. Moreover, continued advancements in edge computing are expected to enable more sophisticated spatial representations and interaction modalities, which will be particularly valuable for extended reality (XR) applications and in-situ systems where low-latency, context-aware processing is essential.

4.1.4 Remarks

The literature reviewed in Chapter 2 highlights that recent advancements in XR technologies have established them as viable and effective interfaces for human-robot collaboration, particularly when integrated with complementary technologies such as digital twins. The conceptual framework proposed in this section – centered around a fully autonomous manipulator with a human-in-the-loop – is intended as an initial step toward achieving more effective integration of robot autonomy and human oversight. By aiming to balance autonomy, operational efficiency, and user involvement, the framework contributes to the broader objective of advancing human-centric cyber-physical systems. Ultimately, the integration of immersive technologies is shown to enhance human control over robotic behavior and situational awareness, thereby supporting the development of adaptive and collaborative industrial systems.

4.2 Virtual Reality for Robot Teleoperation

Section 4.2 describes the implementation setup and experiments conducted for VR-based control of the robot manipulator. The objective is to experimentally evaluate the usability of VR interfaces and immersive environments for enabling safer, more intuitive, and advanced human-robot interaction.

4.2.1 Hardware Setup

The Meta Quest 2, while no longer the latest model in Meta's headset lineup, is employed as the primary virtual reality headset throughout this thesis. It provides the necessary immersive experience for all VR-related experiments conducted in this work. Figure 4.3 provide the visual representation of the HMD and its controllers. Also, a list of specifications of the headset is provided in Table A.2 (Appendix A.2).

Figure 4.3: Meta Quest 2 with Touch Controllers

4.2.2 Software Setup

Game Engines for XR Application

Game engines should be considered as powerful software platforms designed for the creation of video games and various interactive experiences. They provide essential tools for handling user input, physics, AI simulation, and real-time graphics rendering. Popular engines include Unity, Unreal Engine, and Nvidia Isaac Sim offering cross-platform functionality, and enabling developers to build games for consoles, PCs, and mobile devices. The other, perhaps not as widely used engine includes open-sourced Godot. Beyond gaming industry, game engines are also used in fields such as virtual reality, architecture, and film to develop interactive simulations and visual content.

In this thesis the Unity game engine is used as a main development environment for the extended reality applications, featuring the immersive robot teleoperation via

virtual and augmented realities – which is discussed more in detail in the following subsections and the Section 4.3.

Unity Configuration

Unity version 2022.3.45f1 (LTS) is used throughout the whole thesis. This specific version was chosen due to compatibility considerations, as the implementation relies on several third-party plug-ins, some of which are prone to conflicts or lack support in newer Unity versions. The full project repository (XR interactions and the Docker-ROS pipeline from Chapter 3) is available on GitHub[1], which includes the `manifest.json` file detailing all installed packages and their versions. The key packages used in this project are listed below, along with a brief description of their roles:

- OpenXR: Provides a unified interface for developing cross-platform XR applications, ensuring compatibility with the Meta Quest 2 and other OpenXR-compliant devices.
- XR Plugin Management: Manages and loads XR plugins in Unity, enabling the selection and configuration of target XR platforms.
- XR Interaction Toolkit: Supplies components and interaction behaviors for common VR interactions such as grabbing, UI interaction, and teleportation.
- ROS-TCP Connector: Enables communication between Unity and a Robot Operating System (ROS) environment via TCP, facilitating robot control and data exchange.
- URDF Importer: Allows importing of URDF (Unified Robot Description Format) files into Unity, providing visualization and simulation capabilities for robot models.
- Meta Interaction SDK (used in 4.2.5): The collection of the pre-built components and systems that enable natural input methods such as hand tracking, controller input, and eye gaze, along with gesture recognition and ray-based interaction tailored for Meta devices.

[1] https://github.com/ykarpi/Gen3-Immersive-Control

Virtual Interactive Environment

The virtual environment is modeled at a 1:1 scale to accurately replicate the Advanced Control and Intelligent Systems (ACIS) Lab at the University of Victoria, based on the work presented in [10]. It combines assets imported from the Unity Store with custom-developed models to enhance the fidelity of the virtual representation. Figure 4.4 shows an overview of this environment. Figure 4.5 presents an overlay of the first-person view from the Meta Quest 2 headset with a camera image of the corresponding physical space, illustrating the synchronization and visual resemblance between the real and virtual environments.

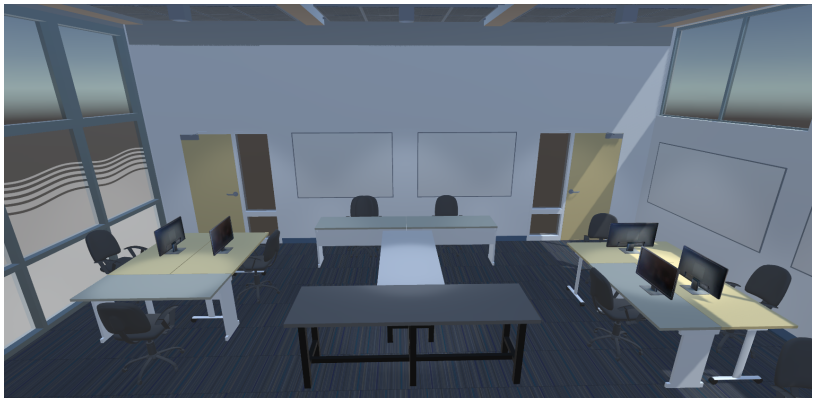


Figure 4.4: Virtual Replica of the ACIS Research Lab at the University of Victoria

Robot Model Import

The URDF Importer[2] in Unity facilitates the seamless integration of robots defined in the Unified Robot Description Format (URDF) into Unity scenes. It parses URDF files to create a corresponding GameObject hierarchy, incorporating components such as Articulation Bodies, joint constraints, inertial properties, and collision meshes. Articulation Bodies, leveraging Unity's PhysX 4.0, enable realistic kinematic and dynamic simulations by defining the physical properties and constraints of robotic joints. Regarding collision management, Unity's physics engine requires that collision meshes be convex to interact correctly with Articulation Bodies. To address this, the URDF Importer employs the Volumetric Hierarchical Approximate Convex Decomposition (VHACD) algorithm, which decomposes complex mesh geometries into convex hulls,

[2]https://github.com/Unity-Technologies/URDF-Importer

which balances computational efficiency with accurate collider behavior for realistic interactions.

Unity-ROS setup

To synchronize, simulate, and ultimately control the physical manipulator (or any robotic system), integration with the Robot Operating System (ROS) is essential. This is achieved using the previously mentioned ROS-TCP Connector[3] package. It enables the Unity project to interface with the ROS network, allowing for both data publication and subscription. A detailed explanation of the ROS and Docker infrastructure is provided in Chapter 3, with specific information on the Unity-ROS integration presented in Subsection 3.2.4.

4.2.3 Implemented Functionality

The immersive control functionality was built upon the ROS network architecture described in Chapter 3. It includes basic commands such as moving the robot to predefined positions, operating the gripper, triggering an emergency stop, and accessing the robot's onboard camera feed, which is projected onto a visual element within the VR environment.

Furthermore, another critical functionality is implemented: manual control of the virtual twin. When the operator activates manual control, the robot's model within the digital twin environment is no longer synchronized with the physical system – essentially turning the digital twin into an offline simulation. This allows the operator to preview potential future robot positions joint by joint and, if satisfied, publish those positions to the physical counterpart. The presented methodology is similar to the designs in [56, 106], allowing the user advanced control.

Also, the accurate virtual environment representation and the implementation's support for simulation tools (such as Gazebo; described in Chapter 3) means that the developed interactions can be used as part of operator training without the presence of the physical system – supporting some of the claims made by the authors in [3] regarding the utility of XR for safe operator training.

[3]https://github.com/Unity-Technologies/ROS-TCP-Connector

4.2.4 User Interface

As part of developing the proof-of-concept for robot teleoperation, a straightforward and simple user interface was created (illustrated in Figure 4.5). Since the main focus is the virtual workbench where the robot model is positioned – accurately reflecting the layout of the physical research lab – the UI windows were placed to the side. This arrangement allows the user to issue commands while keeping the robot's movements in clear view.

Figure 4.5: Real-Time Robot Control via Virtual Reality.

Figure 4.5 presents two perspectives of the same moment: a first-person view from the VR headset and a separate external camera view capturing the physical environment. The virtual user interface elements are taken from the XR Interaction Toolkit, and their design style is kept consistent to maintain clarity and usability in immersive conditions. The implementation uses Meta controllers for interaction with the UI. Additionally, the VR setup allows the operator to navigate the environment in two ways: (1) by physically walking around the lab, since the virtual replica is rendered at a 1:1 scale (as illustrated in Figure 4.5), and (2) by using the controller joysticks to simulate locomotion from a stationary position.

Interactive Joint Control within the Digital Twin Environment

A key functionality is also a possibility of the manual control of the digital twin. When the operator activates manual control, an additional menu appears on the right side of the robot's model (see Figure 4.6), and the virtual robot model decouples from its physical counterpart. This menu consists of six sliders, each corresponding to one of the joints of the 6-DOF Kinova Gen3 manipulator. Adjusting a slider changes the position of the corresponding joint in the virtual robot model, allowing the operator to preview potential movements and configurations, access reachiblity, and safety. Once satisfied, the operator can publish the new joint values, prompting the physical robot to move accordingly. After execution, the digital twin is re-synchronized with the physical system.

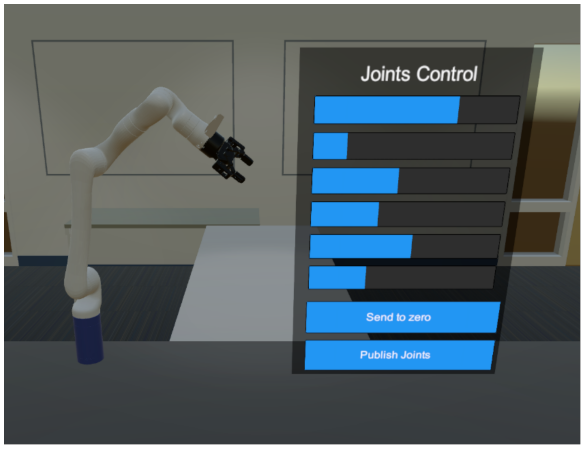


Figure 4.6: UI for Manual Control of Manipulator Joints and Virtual Twin Preview.

4.2.5 Usability Considerations

Immersive technologies present new opportunities for intuitive and effective human-robot interaction. However, the design of user interfaces (UIs) in extended reality (XR) environments remains a nontrivial task, particularly when safety, responsiveness, and clarity are critical, as in the case of robot teleoperation. Usability plays a central role in ensuring the effectiveness of XR applications by minimizing cognitive load, preventing errors, and enhancing overall user experience [109].

To inform the design and evaluation of XR-based robot control interfaces, it is essential to consider established usability principles that promote effectiveness, efficiency, and user satisfaction. Nielsen's 10 Usability Heuristics [76] serve as a foun-

dational framework for assessing interactive systems, offering guidance on visibility of system status, user control and freedom, consistency, error prevention, and other critical aspects. These heuristics are particularly relevant in XR, where interaction is spatial, embodied, and often less constrained than in traditional GUI-based systems.

Revised Interface Design

To improve usability, the revised interface incorporates hand tracking and gesture-based interaction. Two types of hand-based UIs are introduced: one is anchored to the user's palm, providing an always-available, visually consistent menu with grouped buttons (e.g., motion commands, gripper controls, emergency stop); the other is a gesture-activated menu that can be opened and closed by tapping the index finger to the thumb. Both interfaces enable direct interaction via finger taps, resembling touchscreen behavior and eliminating the need for physical controllers. These additions enhance system transparency, improve situational awareness, and support a more intuitive interaction experience. Figure 4.7 illustrates the overall revised UI and interactions.

Since these interface design decisions are made in the context of human-robot interaction, safety is a critical consideration. While the menu includes a dedicated button for emergency stops, it may not always be intuitive or quickly accessible in situations requiring immediate intervention. To address this, a dedicated gesture was integrated to serve as a rapid emergency stop mechanism. This allows the operator to instantly halt the robot's motion and reassess the situation without relying on navigating the interface menu.

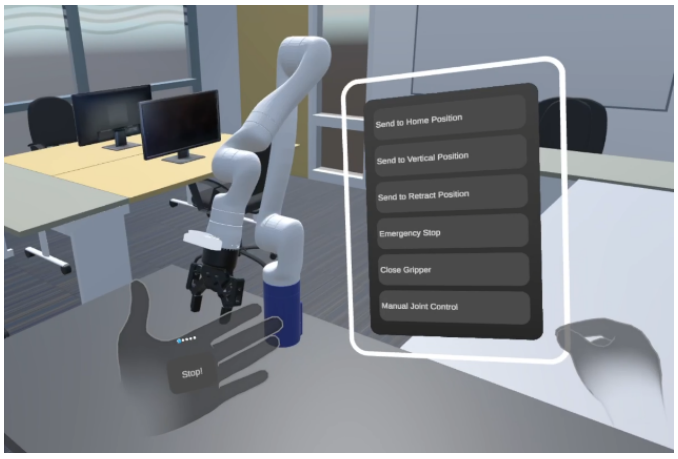


Figure 4.7: Revised UI with Hand Tracking Implementation

The revised interface design incorporates several of Nielsen's usability heuristics, each contributing to a more intuitive and responsive user experience in immersive human-robot interaction. By adopting gesture-based controls that mirror familiar real-world hand movements, such as pinching and tapping, the interface achieves a better "match between system and the real world". This natural mapping reduces the learning curve and helps users feel more in control.

"User control and freedom" is further enhanced through hand-tracked menus that can be summoned or dismissed on demand, enabling flexible interaction and reducing unnecessary visual clutter – a reflection of both "aesthetic and minimalist design" and "flexibility and efficiency of use". To support safety-critical tasks, the integration of a dedicated emergency stop gesture allows for immediate intervention, contributing to the heuristic such as "error prevention".

In addition, the interface aligns with "consistency and standards" by using a coherent visual style, helping users navigate more confidently. While the system currently lacks a dedicated status display, future iterations could benefit from improved "visibility of system status" and mechanisms to help users "recognize and recover from errors", particularly in more complex tasks and interactions.

4.3 From Virtual to Mixed Reality

This section revisits the same use case as in 4.2 – teleoperation of the Kinova Gen 3 robotic manipulator using XR technologies – but shifts the focus toward the transition from virtual reality (VR) to mixed reality (MR). While VR offers a fully immersive environment for system prototyping and operator training, MR introduces unique challenges and opportunities by blending digital content with the physical world. This transition involves additional considerations such as spatial alignment, real-world occlusions, marker tracking and anchoring, and user situational awareness in dynamic environments.

4.3.1 Hardware and Software Setups

Hololens 2: Mixed Reality Headset

The Microsoft HoloLens 2 represents a significant advancement in mixed reality (MR) technology, offering a robust platform for immersive human-computer interaction research. As the second generation of Microsoft's MR headset, it improves on its

predecessor in terms of display resolution, field of view, and particularly hand- and eye-tracking capabilities. The headset is capable of processing complex vision tasks in real time – such as spatial mapping, hand and head tracking. A broad array of input sensors, including RGB and depth cameras, grayscale tracking cameras, and a microphone array, enables high-fidelity environmental understanding and user input [107]. In research contexts, the Research Mode unlocks raw sensor streams for custom processing, making it particularly valuable for prototyping computer vision pipelines, performing multi-modal data collection, and experimenting with on-device algorithms. Empirical evaluations [107, 5] demonstrate strong performance across key metrics such as hand-tracking accuracy, hologram stability, and speech recognition robustness. Its ability to anchor virtual content in physical space and support natural hand and gaze-based interaction makes it suited for mixed reality applications in robotics, training, remote collaboration, and spatial computing [5]. Additional details on the headset's specifications and its comparison to Meta Quest 2 are shown in Table A.2 (Appendix A.2).

Figure 4.8: Mixed Reality Headset – Microsoft Hololens 2

Mixed Reality Toolkit

The Mixed Reality Toolkit 3 (MRTK3) is a cross-platform, open-source development framework developed on top of Unity's XR Interaction Toolkit (XRI) with the aim of facilitating cross-platform extended reality (XR) experience development. The toolkit features a modular, flexible architecture that caters to mixed reality applications on a variety of hardware devices such as the Microsoft HoloLens 2 and Meta Quest headsets. It must be mentioned that although both MRTK2 and MRTK3 are in use, MRTK3 is the newest and most optimized one. This version has been designed specifically to be well compatible with new XR tools and the latest Unity environment. In the context of this project, MRTK3 was selected to ensure better compatibility and smoother integration.

4.3.2 Experimental Design

The developed UI serves as a functional prototype. The functionality as well as the considerations for usability are limited in this design (compared to 4.2.4) with the goal to highlight the differences and considerations that are integral in mixed reality applications development, especially for the robotics or other automation systems.

Figure 4.9 presents the developed control menu, which consists of simple, clearly labeled buttons for sending commands to the Kinova Gen3 manipulator. Taking advantage of the headset's hand tracking functionality, the control menu is implemented as a hand-anchored interface. Unlike VR environments – where spatial coordinates are fully defined and interface placement can be absolute – mixed reality introduces additional complexity due to variable and uncertain environmental grounding. As such, anchoring the interface to the user's hand ensures consistent accessibility and usability. Alternatively, world-anchored menus may be implemented when more reliable spatial grounding is available, as discussed in the following section.

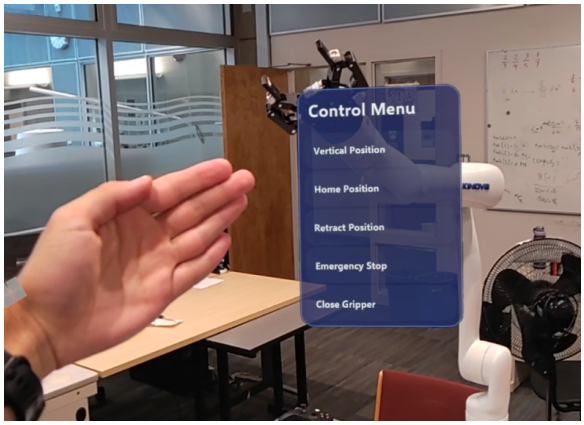


Figure 4.9: Hand Menu for the Robot Teleoperation in Mixed Reality

Multimodal Input

To further explore multimodal interaction, several input methods were implemented and tested. Basic interaction is possible through direct finger tapping or ray-based input. More advanced functionality includes gaze and voice-based interactions. For example, if a user maintains gaze on a button for three continuous seconds, the button is automatically activated. To minimize unintended commands, voice inputs are only recognized when the system also detects the user's gaze focused on the corresponding

UI element. This layered input logic improves both reliability and user experience in dynamic, real-world settings.

World Grounding

One of the practical challenges in working with augmented reality is achieving reliable world grounding. For instance, ensuring that virtual objects appear or disappear at specific real-world locations when the user is looking, or enabling the system to remember where those objects should remain active over time. In the context of this work, world grounding is particularly important for the visualization of the robot model – for example, to accurately preview planned trajectories or to enable in-situ programming.

To support this, image targets can be used – actions are triggered when specific images are detected by the device's camera. While toolkits like Vuforia[4] offer robust image tracking for AR applications, they are not compatible with OpenXR and MRTK3; hence requiring an alternative solution. A viable option is the marker tracking functionality built into OpenXR (via ARFoundation). For instance, QR code detection can be used to anchor virtual content. With appropriate programming, such markers can precisely position objects, such as the robot's virtual model, using the code as a reference point, as illustrated in Figure 4.10.

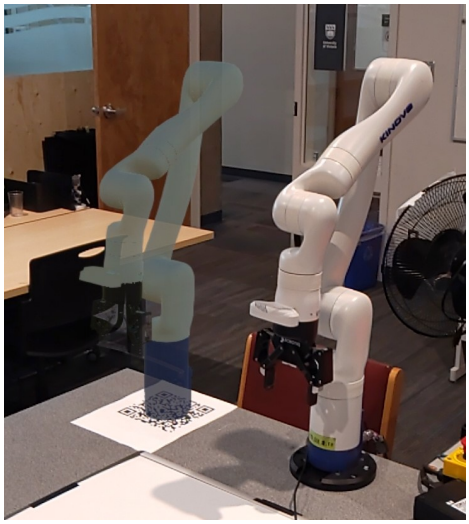


Figure 4.10: Marker Tracking and Robot Visualization in Mixed Reality

[4]https://developer.vuforia.com/

4.4 Discussion: Interaction and Contextual Trade-offs between VR and MR in HRI

The transition from Virtual Reality (VR) to Mixed Reality (MR) introduced a significant shift in how operators interact with the robotic system and their surrounding environment. While both modalities leverage immersive interfaces, they serve different roles in terms of user context, situational awareness, and spatial alignment.

In VR, the operator is fully immersed in a virtual environment, which offers a high degree of control over the visual and interactive design. This allowed for a clean and distraction-free interface to control the robot, preview its movement, and interact with virtual buttons and menus. However, the detachment from the physical world presents its limitations – most notably the loss of spatial context and the inability to see the real robot or workspace during operation.

By contrast, MR overlays the digital interface directly onto the physical environment. This mode of interaction provides enhanced situational awareness, as the operator can see and interact with both the robot and virtual controls simultaneously. Through the MR headset, users are able to place and manipulate 3D UI elements in their surroundings, which is especially useful for tasks requiring accurate spatial understanding or referencing real-world entities. Furthermore, the MR experience can also be extended to mobile devices such as smartphones. While spatial interactions on mobile devices are more limited – particularly due to the absence of full 3D hands, head, motion tracking and gesture recognition – they still support advanced visualizations with basic interaction through touch or device movement.

Based on the conducted review and implementyations, several key distinctions between VR and MR for HRI and general robot control can be drawn::

- Contextual Awareness: VR isolates the user from the physical environment, whereas MR overlays information onto the real world, supporting more context-aware decisions during robot operation.

- Interaction Modality: While both VR and MR support hand tracking, their interaction paradigms differ. VR allows both controller-based and hand-tracking input, with hands typically rendered in the virtual scene. In MR, hand interactions are grounded in the real-world context, often relying on spatial cues and gestures like air tap or direct manipulation anchored to physical space.

- Safety and Feedback: MR provides direct visual feedback of the real robot and environment, which enhances safety and allows real-time monitoring. In VR, all feedback must be simulated, which may lead to delays and mismatches between system state and operator expectations.

In summary, both VR and MR offer distinct yet complementary affordances for immersive robot control. VR provides a fully controlled environment where system behavior, interface layout, and training scenarios can be rigorously designed and evaluated without external distractions. This makes it especially effective for precision tasks, remote operation, or environments where access to the physical system is limited. MR, on the other hand, anchors digital elements in the physical world, enabling real-time interaction with both the robot and its environment – making it ideal for in-situ operation, spatial referencing, and collaborative use cases. The two modes are not mutually exclusive but rather complementary, and future systems could benefit from supporting both depending on task context. This comparison also underscores the value of developing extended reality headsets and potentially the applications capable of seamlessly switching between virtual and mixed reality modes, offering flexibility across a range of industrial and interaction scenarios.

Chapter 5

Realistic Scene Representation for Immersive Applications

A crucial challenge in developing extended reality applications is the creation of interactive environments and virtual elements that enhance immersion. Intelligent spaces and naturalistic interactions are essential for increasing user engagement and realism in virtual and augmented spaces [75]. However, generating 3D models of objects and environments is a time-consuming and resource-intensive process. Therefore, this chapter explores the use of 3D reconstruction techniques – specifically focusing on Gaussian Splatting – for object- and scene-level reconstructions to support immersive applications.

5.1 Background

5.1.1 Novel View Synthesis Techniques

In computer graphics, novel view synthesis refers to the process of generating new images of a scene or subject from arbitrary viewpoints, using only a limited set of images captured from different perspectives. This challenging task has seen significant breakthroughs only in recent years, primarily due to rapid advancements in machine learning. Among the most prominent and effective approaches are Neural Radiance Fields (NeRF) [68] and 3D Gaussian Splatting (3DGS) [52].

Neural Radiance Fields (NeRF) is a groundbreaking method for photorealistic novel view synthesis, which represents a 3D scene as a continuous volumetric field encoded by a neural network. NeRF learns to map spatial coordinates and viewing

directions to emitted color and volume density, enabling it to synthesize highly detailed images from unseen viewpoints. Despite its visual fidelity, NeRF suffers from significant computational overhead due to its reliance on dense sampling and neural inference. These limitations constrain its applicability in real-time applications such as immersive virtual or mixed reality experiences. However, variants like Instant-NGP [73] have improved rendering speed and memory efficiency. Still, NeRF's implicit representation makes it less suited for interactive manipulation and integration with physics or dynamic scene elements.

Three-Dimensional Gaussian Splatting (3DGS) is a novel and effective approach to the real-time scene representation and rendering of 3D scenes, introduced by Kerb et al. [52] as a general-purpose and explicit alternative to volumetric approaches such as NeRF. 3DGS encodes scenes as a collection of anisotropic three-dimensional Gaussians, initialized from sparse point clouds derived from Structure-from-Motion (SfM) [98]. In particular, this capacity enables 3DGS to produce high-quality reconstructions from just Structure from Motion (SfM) data, unlike most point-based methods that require dense Multi-View Stereo (MVS) input. Each Gaussian component of the scene has its position, orientation, spatial covariance, opacity (α), and color – in the form of spherical harmonics (SH) – allowing for a compact yet informative representation of geometry and visual attributes. At training time, an optimization process iteratively updates the Gaussian parameters and introduces adaptive density control to enable the model to be scalable and efficient, typically utilizing 1 to 5 million splats. With its compact size and real-time capabilities, 3D Gaussian Splatting (3DGS) presents itself as a promising solution for use in immersive environments such as virtual and augmented reality, where responsiveness and visual quality are critical.

5.1.2 Facets of Realism in Virtual Environments

Achieving realism within the virtual environments is the question that should be considered and addressed from the perspectives of psychology, human perception, and computer graphics. Realism plays a critical role in fostering immersion within virtual environments, shaping how users perceive and engage with digital spaces. Immersion is defined as the degree to which users feel present within a virtual world, often achieved by aligning visual fidelity, behavioral coherence, and interactive cues to mirror real-world experiences [13, 58]. For instance, the authors in [13] emphasize the importance of coherence across graphical and behavioral realism in maintaining

immersion. Their findings suggest that even significant disruptions in expected behaviors, such as altered gravity or object dynamics, may go unnoticed by players if other immersive elements compensate for the inconsistencies. Behavioral realism, which refers to how closely objects and characters act in comparison to their real-world counterparts, is particularly influential. Cheng and Cairns' [13] study highlights that while graphical changes can elicit initial reactions, the interplay between visual fidelity and expected behavior determines sustained immersion. Similarly, Perroud et al. propose a "realism score" for immersive VR systems, which evaluates how well a system replicates physiological and psychological realism from the perspective of the human visual system [79]. This framework underlines the importance of aligning visual and immersion cues, such as contrast, depth perception, and latency, to optimize user experiences in simulations.

In practice, achieving such realism increasingly relies on advanced 3D reconstruction techniques. Especially in applications where fidelity to real-world spaces is critical – such as training simulations, digital twins, or mixed reality interfaces – 3D reconstruction provides the backbone for visual coherence [55]. Techniques like photogrammetry, depth sensing, and other reconstruction techniques, allow for the creation of detailed and spatially accurate models that closely resemble their physical counterparts. These reconstructions enable accurate spatial alignment, enhance environmental coherence, and increase perceived presence within virtual environments [83]. Studies have shown that high-fidelity reconstructions can lead to stronger affective responses and improved task performance, especially when naturalistic visual and interactive cues are preserved [55]. Furthermore, the use of accessible hardware (e.g., smartphones) and automated reconstruction pipelines democratizes the creation of high-quality digital twins, making immersive and realistic VR experiences more widely attainable [59, 55].

5.1.3 3DGS Integration with Game Engine and XR

The subsection 4.2.2 has made an introduction of the game engines, and their utility for development of the immersive and interactive experiments. The Unity engine is used as a primary development platform in this chapter (in this thesis in general), it is a free to use with multiple plugins and toolkits for XR, and it's the most used platform in the existent literature among the academic community.

One of the first plugins for importing Gaussian splats into Unity, developed by

Aras Pranckevičiu [85], allows users to import `.ply` files and compress splats to adjust quality and asset size. The pipeline renders splats after opaque objects and the skybox for proper occlusion with the Z-buffer, but since they don't write to the Z-buffer, splats are rendered before transparent objects and don't interact well with semi-transparent materials, causing potential rendering issues. The plugin also enables manual editing of splats, including making cutouts, and supports combining multiple GS-rendering game objects for further editing or exporting back into `.ply` format. Although the work in [85] demonstrates a successful integration with Unity, initially it was not specifically optimized for extended reality applications. In contrast, the authors in [16] implemented Differential Gaussian Rasterization as a Unity native plugin tailored for XR use, enabling the viewing of GS models. This approach has been featured in several papers [86, 47]. Another plugin built on [85], optimized for XR, is described in [54], showcasing a multi-layer GS experience for human anatomy visualization. Both implementations are compatible with Unity's XR Management plugin and OpenXR, which are now the standard. All of the above-mentioned projects are open-sourced and available for developers, enthusiasts, and researchers.

5.2 Object-level Reconstruction: Preliminary Investigation

Recent advancements in 3D reconstruction technologies have significantly lowered the barrier to entry for capturing high-fidelity digital environments. Several companies have emerged as leaders in this space, providing tools that allow users - even those without specialized hardware - to scan and reconstruct physical spaces using only mobile devices. This growing accessibility is especially valuable for immersive technology applications, where cost, portability, and ease of integration are critical. Polycam[1] has gained popularity for its mobile app that leverages LiDAR and photogrammetry, enabling the creation of detailed 3D models directly from smartphones and tablets. It is widely used across industries such as architecture, gaming, and the arts. LumaLabs[2] has pushed the envelope with AI-driven reconstruction techniques like Neural Radiance Fields (NeRF) and Gaussian Splatting, allowing photorealistic 3D scene reconstruction from just a few images. Niantic[3], best known for AR applications like

[1] https://poly.cam
[2] https://lumalabs.ai
[3] https://nianticlabs.com

Pokémon GO, also actively developes tools for mobile 3D mapping. Among its products is Scaniverse, a smartphone application that allows users to capture 3D scans of real-world environments and objects. Designed with everyday users in mind, Scaniverse makes it possible to create highly detailed reconstructions without requiring specialized scanning equipment.

As part of the preliminary experiments, Scaniverse was selected for 3D object capture due to its ease of use, compatibility with mobile hardware, and support for advanced rendering techniques such as Gaussian Splatting. The purpose of this study is to explore how readily accessible consumer technology can be leveraged to generate 3D assets for use within extended reality (XR) environments.

5.2.1 Case Study: Rubik's Cube

The Rubik's Cube was selected as the subject of a detailed case study focused on data capture, processing, and integration with the Unity engine for XR interactions. Its well-defined geometric structure – featuring flat, uniformly sized faces and right angles – makes it potentially an ideal candidate for evaluating the accuracy and fidelity of 3D reconstruction methods. The cube's multicolored, high-contrast surface patterns further support the assessment of texture mapping and color reconstruction. In the context of virtual environments, the Rubik's Cube serves as a recognizable and compact object that allows for precise interaction testing, offering clear visual feedback when manipulated in XR.

Methodology

Figure 5.1 presents an overview of the methodology pipeline: the data is first captured and processed using the Scaniverse application, then the resulting reconstruction is exported and refined within the Supersplat[4] editor (allowing to get rid of the splats that do not belong to the object of interest); finally, the edited model is imported into the Unity scene for use in the XR application. After generating the Gaussian Splatting-based representation of the Rubik's Cube, the resulting `.ply` file was edited in the Supersplat playcanvas editor to isolate the cube from the surrounding scene captured during the scanning process. This refinement step was performed manually, introducing a degree of subjectivity that may influence some of the assessment. Figure 5.2 shows the side views of the reconstructed object. Although, it has uniform and

[4]https://superspl.at/editor

flat surfaces, some minor imperfections are noticeable even after meticulous manual editing.

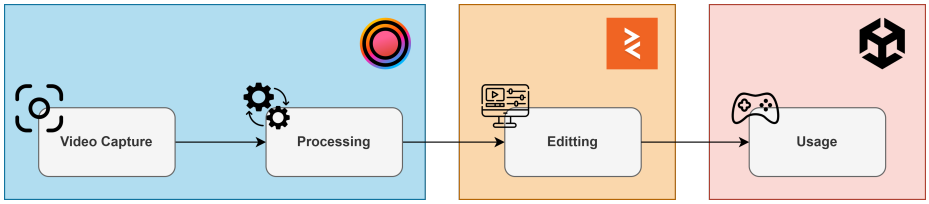


Figure 5.1: Overview of the Data Processing Pipeline: 3D Scanning, Followed by Point Cloud Refinement, and Game Engine Integration

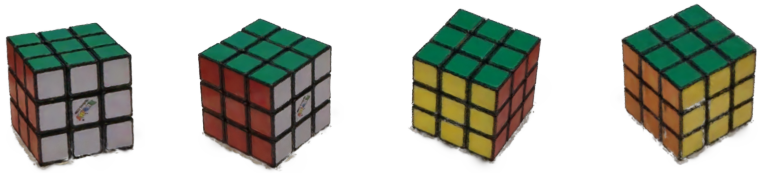


Figure 5.2: Side Views of the Reconstructed Rubik's Cube Using Gaussian Splatting

One of the limitations of 3D reconstruction methods, applicable not only to Gaussian Splatting is the inability to capture surfaces that are in direct contact with another object, such as the ground or a table. In this case, the bottom face of the Rubik's Cube, which was resting on the table during scanning, was occluded and therefore not reconstructed (which is a common challenge in single-pass 3D scanning).

5.2.2 Visual Representation and Interactions

After the `.ply` file has been edited in the Supersplat editor to isolate the Rubik's Cube from the surrounding environment, it is imported into the Unity scene for further integration (the methodology for this process is detailed in Section 5.1.3). However, one of the key limitations of Gaussian Splatting-based representations becomes apparent at this stage. Unlike traditional 3D reconstruction methods that generate polygonal meshes, GS produces point-based renderings that lack an underlying mesh structure. This poses a significant challenge since most XR software development kits (SDKs) and game engines, including Unity, rely heavily on meshes to interpret and interact with the virtual environment.

To enable interaction with the GS-based Rubik's Cube in Unity, key components are added to the imported object. The `Object Manipulator` allows users to grab,

move, and rotate the object using input devices or hand gestures. Since GS lacks physical boundaries or mesh topology, a `Box Collider` approximates the object's volume for collision detection. A `Rigidbody` is added to integrate with Unity's physics engine, enabling natural responses to forces and gravity. These components collectively bridge the visual realism of GS with physical interactability.

Furthermore, to validate the GS-based methodology, the same Rubik's Cube was also reconstructed using a Photogrammetry approach (also using the pipeline similar to Fig. 5.1 using Scaniverse capabilities) and imported into the Unity scene alongside a CAD model serving as a reference. Figure 5.3 illustrates all three objects on a surface within the virtual environment, while Figure 5.4 showcases interactions with the 3DGS-based asset model in both virtual and mixed realities on different XR platforms.

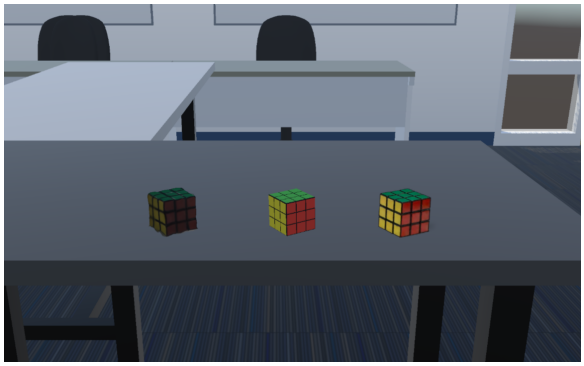
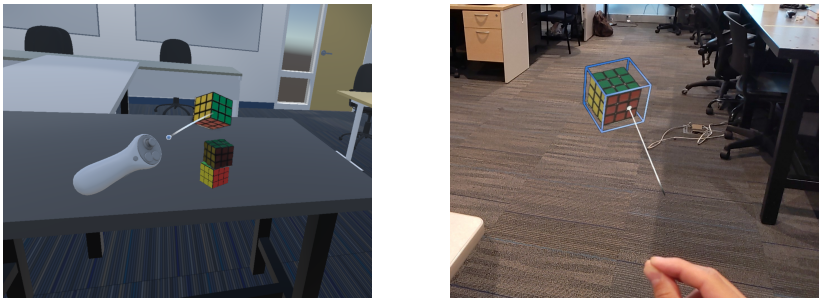


Figure 5.3: Three Rubik's Cubes in the Virtual Reality Environment: Photogrammetry-Based (Left), CAD Model (Center), Gaussian Splat-Based (Right)



(a) VR (with Meta Quest 2) (b) MR (with Hololens 2)

Figure 5.4: Interaction with Reconstructed Objects in Virtual and Mixed Realities

Remarks

The Rubik's Cube as the study object was represented in the virtual environment using three different methods: a CAD model, a 3D Gaussian Splatting based model, and a photogrammetry-based model. Each method offers distinct advantages and limitations in terms of visual quality, realism, and compatibility with interactive experiences in XR. The CAD model is used as a reference, providing a highly accurate geometric representation. In VR (Fig. 5.3), this model appears clean and fully suitable for use cases requiring precision, collision-based interactions, or simulation of mechanical behavior. However, it lacks surface texture variation and realism – the visual fidelity is limited to preset colors or manually applied materials, which often results in a somewhat artificial appearance.

The 3DGS-based model delivers photorealistic rendering quality, capturing subtle surface details and appearing impressively lifelike in VR. However, due to its view-dependent nature, 3DGS does not provide a solid geometric structure – after all, it consists of a collection of Gaussians rather than defined surfaces. This limits its applicability in interaction scenarios, as it becomes difficult to enable precise collision detection or physically manipulate the object like traditional mesh-based models. Despite these limitations, 3DGS excels in visual fidelity and immersive realism, particularly for passive visualization.

The photogrammetry-based Rubik's Cube represents a middle ground. It offers realistic texture and surface detail, including some imperfections visible on the object. Unlike 3DGS, it produces an actual 3D mesh, which allows for significantly more straightforward integration with physics systems. Additionally, like the CAD model, it responds to dynamic lighting in the scene, creating highlights and shadows that enhance spatial perception. However, the geometry is quite noisy or less precise than the CAD model, and visual artifacts are present around complex edges or in occluded areas. In VR, this model might be potentially suitable for scenarios that balance realism with interactivity.

5.2.3 Quantitative Analysis and Metrics

To quantitatively support the analysis of the 3D Gaussian Splatting (3DGS) reconstruction and to facilitate a direct comparison with a traditional photogrammetry-based method, the axis-aligned bounding dimensions of the resulting point clouds were evaluated. Although both reconstruction pipelines rely on Structure-from-Motion

(SfM) as an initial step, the resulting representations differ in structure: 3DGS produces a point-based volumetric radiance field represented by anisotropic Gaussians, while photogrammetry typically outputs a dense textured mesh.

Data Preparation

To enable point-based comparison, the photogrammetry-derived mesh was converted into a point cloud via uniform surface sampling. A total of 50 000 points were sampled, which proved sufficient; increasing the sampling rate to 100 000 or 200 000 points altered the computed bounding box dimensions by less than 0.01%.

Since the presence of outliers in a point cloud can disproportionately affect geometric measurements, a preprocessing step was employed to improve robustness for both reconstruction methods. A statistical outlier removal [91] technique was applied based on local point density. Specifically, for each point, the average distance to its k nearest neighbors (k=*20*) was calculated. Points whose mean neighbor distance exceeded two standard deviations from the global mean were considered as outliers and removed. This cleaned subset was then used to compute the object's axis-aligned bounding box (AABB), yielding a more reliable and interpretable estimate of spatial dimensions for comparison.

Results

Table 5.1 shows a summarized overview of the obtained dimensions for the point clouds, and their comparisons to the object's original size. Metrics such as absolute and relative differences, MAPE (%), and euclidean distance were used as part of the comparison. Section A.3 in Appendix A provides some additional information on the mathematical foundation of the presented metrics.

The comparison highlights subtle differences in geometric accuracy between the CAD model, 3D Gaussian Splatting (GS), and photogrammetry-based reconstructions. While both GS and photogrammetry approximate the original object's dimensions reasonably well, each shows axis-specific deviations – GS tends to slightly overestimate overall size. On average, photogrammetry achieves closer alignment, as reflected in its marginally lower error metrics. These findings suggest that although neither method perfectly replicates the original geometry, both offer practical representations depending on the trade-off between accuracy and visual realism. It is also worth noting that both reconstructions were created using methods accessible to

Table 5.1: Axis-Aligned Size and Geometric Differences Between Original Object and Reconstructed Representations.

Source	Metric	x	y	z
Physical object / CAD	Bounding Dimensions [cm]	5.75	5.75	5.75
Gaussian Splatting	Bounding Dimensions [cm]	6.12	6.20	6.24
Photogrammetry	Bounding Dimensions [cm]	6.33	5.77	6.20
Original vs GS	Abs. Diff. [cm]	0.375	0.451	0.491
	Rel. Diff. [%]	6.52	7.85	8.54
MAPE [%]			7.63	
Euclidean Dist. [cm]			0.765	
Original vs Photogrammetry	Abs. Diff. [cm]	0.583	0.016	0.447
	Rel. Diff. [%]	10.14	0.28	7.78
MAPE [%]			6.07	
Euclidean Dist. [cm]			0.735	

everyday users and included a manual data cleaning step, which may have influenced the final bounding box measurements.

5.3 Scene-level reconstruction: Research Lab

The goal of this part of the project was to represent the ACIS Lab at the University of Victoria using the Gaussian Splatting (GS) reconstruction method. For instance, Figure 4.4 showcases the lab environment constructed using a combination of custom-made CAD models and prebuilt Unity assets. While the figure depicts a detailed virtual environment – accurately modeled at a 1:1 scale and including major elements such as desks, desktop computers, windows, and other essential furnishings – it is still clearly artificial. Notably, it lacks finer details such as cables, personal items, or subtle surface textures that contribute to realism. This motivated the exploration of GS-based reconstruction as a means to capture and render real-world complexity with higher visual fidelity. In this section, the 3D reconstruction was carried out using a different approach: the Gaussian Splatting codebase was installed and run locally on a PC. After capturing the data using a smartphone's camera, the reconstruction and optimization process was performed entirely on the local machine, without relying on any third-party services or cloud-based tools.

It should be noted that this part of the work does not focus on optimizing the

internal components of the GS codebase. Instead, it explores how this technology can enhance the realism of virtual environments and interactions in XR, including its potential benefits for robot teleoperation.

5.3.1 Installation and Setup

All of the data preparation steps as well as optimization code was run on a Windows 10 machine. Although the original Gaussian Splatting code is made for Linux, there have been multiple implementations that adopted it for a Windows platform[5] and available for others to replicate.

The overall process can be divided into three parts: data preparation, optimization (i.e., training), and visualization of the resulting splat-based rendering. The data preparation stage begins with capturing a video walkthrough of the ACIS Lab to cover the scene from multiple angles. Frames are then extracted from this video to serve as input images for the reconstruction process. Multi-view frames are processed using COLMAP to extract camera poses and reconstruct the scene geometry. This data initializes the Gaussian Splatting optimization, which refines the parameters of thousands of 3D Gaussians to produce a photorealistic representation of the environment.

Multiple experiments were carried out using various video lengths, recording modes, and camera perspectives. The following section provides a detailed description of the final experiment, which produced the highest rendering quality among all trials:

- The video was recorded at Full HD resolution (1920×1080) and 60 fps.
- The total video duration was 52.5 seconds. The camera was moved around the robot workstation in an object-centric manner. To generate the dataset, 3 frames were extracted per second, resulting in a total of 158 images.

5.3.2 Reconstruction Analysis

After running the training (optimization) script, the resulting reconstruction was saved as a `.ply` file. Figure 5.5 shows the output from a zoomed-out viewpoint that reveals the overall structure and the gaps in the reconstruction. As shown in the

[5]https://github.com/jonstephens85/gaussian-splatting-Windows

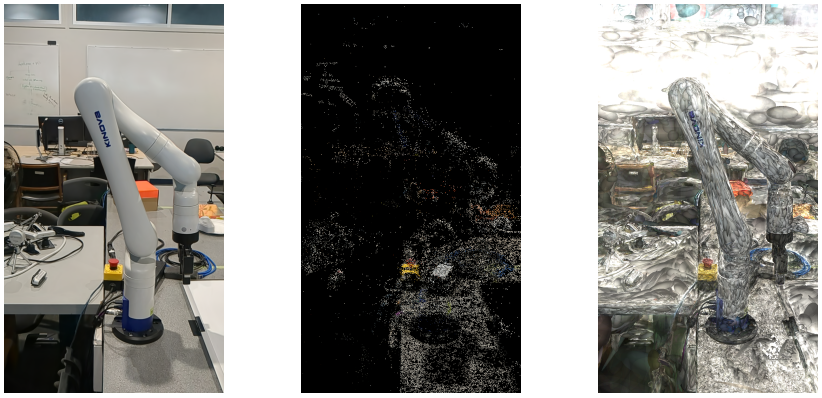
figure, the reconstruction contains multiple visible gaps. This reflects one of the key characteristics – and limitations – of the original GS codebase: it is primarily designed for object-centric views. The video used for reconstruction was indeed object-centric, with the robot manipulator as the focal point. However, even in this setup, the user's physical movement was restricted by desks and equipment in the lab, making it difficult to capture areas such as the floor, ceiling, and finer details of objects located farther from the robot workstation.

Figure 5.5: Zoomed-Out View of the Reconstructed Research Lab

Attempts were made to move beyond the object-centric capture approach by recording video while moving more freely around the research lab. However, it became clear – immediately after running the COLMAP feature matching script – that this approach was unsuccessful. The reconstruction failed to proceed, primarily due to inaccurate or incomplete camera pose estimation. Without accurate camera poses, the Gaussian Splatting optimization process cannot converge to a coherent 3D representation of the scene. That said, recent developments in the literature have explored more open-scene reconstructions. For instance, Hierarchical Gaussian Splatting [53] introduces a pipeline designed for large-scale datasets, such as those captured from vehicle-mounted cameras. Exploring such methods posits a promising direction for future work.

Although the reconstruction in Figure 5.5 appears to contain gaps, it represents a zoomed-out overview of the entire scene. When the virtual camera is aligned with, or positioned close to, the original camera poses used during capture, the quality of the

reconstruction improves drastically. This suggests that while global completeness may be limited, localized fidelity near the original viewpoints remains high, and sometimes it might be tricky to distinguish whether a certain view is a captured image or a reconstruction (e.g. Figure 5.6a).



(a) Splats (b) Initial Points (c) Ellipsoid

Figure 5.6: Render Modes of Gaussian Splatting Reconstruction

Figure 5.6 demonstrates several of these modes: splat-based rendering (left), the raw 3D point cloud (center), and the ellipsoid representation (right). Each rendering mode reveals different levels of the underlying scene representation, offering valuable insights into the structure and density of the reconstructed data.

5.3.3 Unity Integration and Robot Model Representation

3D reconstruction provides a visually accurate and appealing representation of the environment, and this also includes the model of the manipulator. After running the GS optimization for our scene, the robot model is reconstructed as it is located in the lab (the workbench in the center in Fig. 5.5). In comparison to the robot's representation described in the VR and AR interaction sections (Chapter 4), this model, though visually accurate, is not functional.

It is worth mentioning that recent research, such as presented in [61], has explored the use of segmentation models to identify and isolate each joint and link of a robot within a Gaussian Splatting (GS) reconstruction. This segmentation can then be used to assign Denavit-Hartenberg (DH) parameters to the robot, enabling accurate mechanical behavior. While this approach shows significant potential for future work, it is outside of the scope of this thesis work, therefore, a different method was adopted.

Specifically, after importing the splats into the Unity environment, built-in editing tools were used to manually create cubic masks to remove splats corresponding to the Kinova Gen3 robot. The robot's URDF file was then imported and positioned in its place. Although this process requires considerable manual adjustment, it effectively serves as a prototype to validate the concept. In fact, the methodology adopted in this work was inspired by Bowser et al. [9], who demonstrated the reconstruction of a synthetic scene with the virtual robot model, and subsequently focused on aspects of human-robot interaction within such environment.

Figure 5.7 shows the in-game view as seen by the user in a VR headset, divided into three parts: (a) the original splat-based rendering, (b) the scene view with the splats corresponding to the Kinova Gen 3 robot removed – revealing the background while omitting the robot itself, and (c) a carefully aligned URDF model of the robot positioned in place of the removed splats.

(a) Splat-based (b) Cutout (c) URDF-based

Figure 5.7: Robot Representation within the 3DGS-based Environment in VR.

5.4 Discussion

In the context of immersive technologies, 3D Gaussian Splatting (3DGS) enables photorealistic rendering that can significantly enhance a user's sense of presence and reduce cybersickness [39]. However, despite these visual advantages, most existing studies have tested 3DGS in controlled or limited settings and have yet to address several practical challenges critical for immersive and interactive applications, such as physics-like behavior, dynamic interaction, and lighting coherence.

One of the core benefits and limitations lies in the explicit representation. It allows for real-time rendering and provides significantly more interpretability into the reconstruction process, in comparison to, for example, NeRF's black-box nature. However,

unlike traditional 3D reconstruction pipelines (like photogrammetry) that produce polygonal meshes, 3DGS relies on a point-based representation, which lacks topological structure. This becomes a significant bottleneck in XR development, where mesh-based geometry is essential for enabling physics-based interactions, collision detection, haptic feedback, and spatial anchoring. Most XR software development kits (SDKs) and game engines, such as Unity or Unreal, heavily rely on mesh-based representations, which limits the direct integration of Gaussian splats into interaction-rich XR scenarios.

Several recent efforts have explored mesh extraction from 3DGS renderings to bridge this gap [35, 110, 115]. However, these conversions often yield dense, noisy, and topologically unreliable meshes that are not optimized for real-time rendering or interaction. Furthermore, the physical properties of materials – such as rigidity, reflectivity, or mass – are not inherently modeled in 3DGS. These must either be manually assigned or inferred with the help of machine learning techniques, as explored in works like [47, 64, 116]. Even advanced techniques for lighting and reflection handling, such as relightable Gaussians [30] or 3DGS ray tracing [70], remain largely untested in real-time XR environments. In response to these limitations, hybrid approaches are emerging. For instance, Dongye et al. [24] propose a probabilistic framework that dynamically chooses between mesh- or Gaussian-based object representation depending on the user intent in real-time virtual reality setting.

3DGS finds its application in robotics for various tasks, such as guidance [125], sim2real, and even real2sim2real [61]. However, its direct applicability and testing in the domain of human-robot interaction (HRI) remain limited, primarily due to the limitations mentioned above. Nonetheless, the methodology's photorealism can significantly improve situational awareness and contextual understanding (as shown by [9]), a benefit that is particularly useful for operator training or rapid response teams to obtain a more accurate, lifelike picture of the environment. Overall, 3DGS's ability to capture and render real-world scenes and objects with high visual fidelity remains vital. Such capabilities lead to blurring the lines between physical and digital, thereby contributing to further evolution of human-centered cyber-physical systems.

Chapter 6

Concluding Remarks and Future Outlook

The work presented in this thesis highlights the complexity and multi-disciplinary nature of the human-robot interaction (HRI) field. Designing effective collaborative systems within this domain requires a careful consideration of numerous aspects, ranging from the robotics setup and the communication modalities to the enabling devices, technologies, and interfaces that bridge physical and digital spaces. This research demonstrates how integrating spatial computing and digital twin frameworks can enhance human-robot collaboration by promoting more natural, intuitive, and context-aware interactions. These advancements aim to support more human-centered cyber-physical systems capable of operating within complex, evolving industrial contexts.

Building toward this goal, the thesis introduced a modular and scalable framework that combines containerized robotic systems with immersive user interfaces. A Docker-based architecture for ROS was used to support platform-agnostic deployment and development of the robotic application. Building on this infrastructure, the thesis explored virtual and mixed reality implementations for teleoperation and in-situ robot interactions. A human-in-the-loop framework was also conceptualized, highlighting XR's critical role in conveying operator intent, supporting decision-making, and involving users in the robot learning process. Finally, the investigation of realistic scene representation techniques, primarily focused on 3D Gaussian Splatting, examined how visual realism and spatial fidelity affect user perception and interaction within immersive environments. Together, these components demonstrate a multi-

faceted approach to designing and deploying immersive HRI systems that are both technically robust and user-centered.

One of the key insights of this work is that the effectiveness of immersive human-robot systems hinges as much on their design and integration as on their technical capabilities. The integration of digital twin frameworks with real-time data flow and processing enables more intelligent, responsive, and adaptive robotic behavior. The comparative exploration of VR and MR modalities highlights how immersive technologies can be tailored for different operational needs. For example, virtual reality is particularly well suited for simulation, remote control, and training. Meanwhile, mixed reality is crucial for real-world in-situ interactions where situational awareness is of utmost importance. Across both, user-centered design remains a critical factor to ensure transparency, usability, and safety. Importantly, the findings reinforce the broader vision of human-centered cyber-physical systems, where human ingenuity combines with machine precision and autonomy. This enables more flexible and resilient processes across diverse real-world applications, from traditional industrial and manufacturing to cognitive robotics, healthcare, logistics, and assistive technologies.

Moving forward, this thesis lays the groundwork for enhancing human-robot interaction and implementing human-centered cyber-physical systems by leveraging spatial technologies, while also identifying key use cases and practical challenges. Building on this foundation, several promising research directions emerge at the intersection of robotics, extended reality (XR), and artificial intelligence (AI). One such direction is the envisioned industrial metaverse, where real-time digital twins, immersive environments, and collaborative AI agents converge to support operational control, training, and decision-making. Continued research is needed to support transitions between physical and digital workflows, where spatial computing not only visualizes information but serves as a primary medium of interaction with real-world systems.

A central theme in this evolution is the ongoing blurring of boundaries between physical and digital spaces, which opens new opportunities for enhancing human-robot interaction. Future HRI systems should further explore how embodied presence, spatial awareness, and contextual feedback in immersive environments affect user perception, trust, and decision-making. In particular, psychological factors, such as how users interpret a robot's motion, behavior, or apparent intent, will become increasingly important in the design of interfaces that feel natural and safe. Another critical direction involves the integration of large language models (LLMs) into immersive systems. Treating robots as semi-autonomous, NPC-like agents capable of

contextual dialogue could fundamentally reshape HRI. By grounding LLMs in spatial context, for example, via visual input, scene semantics, or task memory, such agents could provide real-time guidance, clarification, and adaptive interaction in collaborative settings. However, this vision also raises challenges around social expectations, conversational nuance, and ethical transparency, which must be addressed through new design approaches. More broadly, AI-driven systems that span perception, planning, and interaction will play a transformative role in next-generation HRI. Aligning high-level human intent with low-level robotic control, dynamically interpreting task context, and tailoring interfaces to individual users all contribute to building intelligent and adaptive human-centered cyber-physical systems.

Finally, advancements in edge computing and XR hardware are expected to further accelerate the adoption of immersive HRI and popularization of XR technology. As wearable XR devices become lighter, more ergonomic, and capable of on-device inference, the feasibility of deploying them in real-world industrial contexts will grow. Reduced latency and better integration into daily workflows will enable mobile, untethered interaction across factory floors, training facilities, and remote operation centers. While this may not be specific only to robotics and HRI, such infrastructure is a key enabler of seamless, spatially aware human-machine interfaces.

Appendix A

Additional Information

The sections in the Appendix A are independent of one another; each provides supplementary information to support the analysis or offer additional contextual details relevant to the main chapters of the thesis.

A.1 Nielsen's Usability Heuristics

To support the discussion in Section 4.2.5, this part of appendix provides an overview of Nielsen's 10 Usability Heuristics for Interface Design. These principles serve as widely adopted guidelines in the field of human-computer interaction, including the design of XR interfaces.

Table A.1: Nielsen's 10 Usability Heuristics for Interface Design

#	**Heuristic**
1	Visibility of system status
2	Match between system and the real world
3	User control and freedom
4	Consistency and standards
5	Error prevention
6	Recognition rather than recall
7	Flexibility and efficiency of use
8	Aesthetic and minimalist design
9	Help users recognize, diagnose, and recover from errors
10	Help and documentation

A.2 Head-Mounted Display Specifications

To support the implementations discussed in Chapter 4, the following table provides an overview of the specifications for the virtual reality (VR) and mixed reality (MR) headsets used in this work.

Table A.2: Technical Specifications of Meta Quest 2 and HoloLens 2

Component	**Meta Quest 2**	**HoloLens 2**
Processor	Snapdragon XR2	Snapdragon 850 + HPU 2.0
RAM	6 GB	4 GB LPDDR4x
Storage	128 GB	64 GB UFS 2.1
Display	Fast-switch LCD	See-through holographic lenses (waveguide)
Resolution (per eye)	1832 × 1920 px	2048 × 1080 px
Refresh Rate	Up to 120 Hz	60 Hz
Field of View	97° horizontal	52° diagonal
IPD Adjustment	3 settings (58–68 mm)	Automatic
Tracking	Inside-out 6DoF	Inside-out 6DoF with eye, head, and hand tracking
Controllers	Oculus Touch (Gen 3)	None (fully hand-tracked)
Hand Tracking	Supported	Fully supported
Eye Tracking	Not supported	Supported
Audio	Built-in speakers & mic	Spatial sound with built-in speakers & mic array
Ports	USB-C, 3.5 mm audio jack	USB-C
Wireless	Wi-Fi 6, Bluetooth 5.0	Wi-Fi 5 (802.11ac), Bluetooth 5.0
Battery Life	2–3 hours	2–3 hours
Weight	503 g	566 g

A.3 Formal Description of SOR Filtering and Dimensional Metrics

The sections provides mathematical foundations for the outliers removal and used metrics in Chapter 5 for the quantitative evaluation of the reconstructed object.

A.3.1 Statistical Outlier Removal with K-NN

Let $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$ be a point cloud of N points, where each $p_i \in \mathbb{R}^3$.

1. Compute Mean Distance to k Nearest Neighbors:

   For each point p_i, compute the average Euclidean distance to its k-nearest neighbors:

   $$d_i = \frac{1}{k} \sum_{j=1}^{k} \|p_i - \mathrm{NN}_j(p_i)\| \tag{A.1}$$

   where $\mathrm{NN}_j(p_i)$ is the j-th nearest neighbor of p_i, and $\|\cdot\|$ denotes the Euclidean norm.

2. Compute Global Statistics:

   Compute the global mean μ_d and standard deviation σ_d of all d_i:

   $$\mu_d = \frac{1}{N} \sum_{i=1}^{N} d_i \tag{A.2}$$

   $$\sigma_d = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_i - \mu_d)^2} \tag{A.3}$$

3. Identify and Remove Outliers:

   Define a threshold $T = \mu_d + \alpha \cdot \sigma_d$, where α is a chosen constant (e.g., 2). A point p_i is considered an outlier if:

   $$d_i > T \tag{A.4}$$

   The filtered point cloud $\mathcal{P}' \subset \mathcal{P}$ contains only points for which $d_i \leq T$.

A.3.2 Quantitative Evaluation Metrics

To quantify geometric differences between reconstructed point clouds, the following metrics were computed based on axis-aligned bounding box (AABB) dimensions:

Absolute Difference (Abs. Diff.)

The absolute difference for each axis $i \in \{x, y, z\}$ is defined as:

$$\Delta_i^{\text{abs}} = \left| D_i^{(r)} - D_i^{(o)} \right| \tag{A.5}$$

where:

- $D_i^{(r)}$ is the reconstructed bounding box dimension,
- $D_i^{(o)}$ is the original (reference) bounding box dimension.

Relative Difference (Rel. Diff.)

The relative difference for each axis i is computed as:

$$\Delta_i^{\text{rel}} = \frac{\left| D_i^{(r)} - D_i^{(o)} \right|}{D_i^{(o)}} \times 100\% \tag{A.6}$$

Mean Absolute Percentage Error (MAPE)

To summarize the relative differences across all axes, the Mean Absolute Percentage Error is calculated as:

$$\text{MAPE} = \frac{1}{3} \sum_{i \in \{x,y,z\}} \frac{\left| D_i^{(r)} - D_i^{(o)} \right|}{D_i^{(o)}} \times 100\% \tag{A.7}$$

Euclidean Distance

The Euclidean distance quantifies the overall geometric discrepancy between the AABB dimensions in 3D space:

$$d_E = \sqrt{\sum_{i \in \{x,y,z\}} \left(D_i^{(r)} - D_i^{(o)} \right)^2} \tag{A.8}$$

This metric reflects the magnitude of dimensional deviation.

Bibliography

[1] J. Allspaw, G. LeMasurier, and H. Yanco. Comparing performance between different implementations of ros for unity. 2023.

[2] J. Arents and M. Greitans. Smart industrial robot control trends, challenges and opportunities within manufacturing. *Applied Sciences*, 12(2):937, 2022.

[3] S. B. i. Badia, P. A. Silva, D. Branco, A. Pinto, C. Carvalho, P. Menezes, J. Almeida, and A. Pilacinski. Virtual reality for safe testing and development in collaborative robotics: challenges and perspectives. *Electronics*, 11(11):1726, 2022.

[4] R. Baheti and H. Gill. Cyber-physical systems. *The impact of control technology*, 12(1):161–166, 2011.

[5] P. Balakrishnan and H.-J. Guo. Hololens 2 technical evaluation as mixed reality guide. In *International Conference on Human-Computer Interaction*, pages 145–165. Springer, 2024.

[6] O. Bentaleb, A. S. Belloum, A. Sebaa, and A. El-Maouhab. Containerization technologies: Taxonomies, applications and challenges. *The Journal of Supercomputing*, 78(1):1144–1181, 2022.

[7] L. P. Berg and J. M. Vance. Industry use of virtual reality in product design and manufacturing: a survey. *Virtual reality*, 21:1–17, 2017.

[8] G. Bolano, Y. Fu, A. Roennau, and R. Dillmann. Deploying multi-modal communication using augmented reality in a shared workspace. In *2021 18th International Conference on Ubiquitous Robots (UR)*, pages 302–307. IEEE, 2021.

[9] S. Bowser and S. M. Lukin. 3d gaussian splatting for human-robot interaction. In *The 1st InterAI Workshop: Interactive AI for Human-centered Robotics*, 2024.

[10] T. Charter. Human-centered intelligent monitoring and control of industrial systems: A framework for immersive cyber-physical systems, 2024.

[11] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin. Smart factory of industry 4.0: Key technologies, application case, and challenges. *Ieee Access*, 6:6505–6519, 2017.

[12] H. Chen, M. C. Leu, and Z. Yin. Real-time multi-modal human–robot collaboration using gestures and speech. *Journal of Manufacturing Science and Engineering*, 144(10):101007, 2022.

[13] K. Cheng and P. A. Cairns. Behaviour, realism and immersion in games. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1272–1275, 2005.

[14] S. H. Choi, K.-B. Park, D. H. Roh, J. Y. Lee, M. Mohammed, Y. Ghasemi, and H. Jeong. An integrated mixed reality system for safety-aware human-robot collaboration using deep learning and digital twin generation. *Robotics and Computer-Integrated Manufacturing*, 73:102258, 2022.

[15] C.-H. Chu and Y.-L. Liu. Augmented reality user interface design and experimental evaluation for human-robot collaborative assembly. *Journal of Manufacturing Systems*, 68:313–324, 2023.

[16] Clarte. A vr viewer for gaussian splatting models in unity, April 2024.

[17] Y. E. Cogurcu and S. Maddock. Augmented reality safety zone configurations in human-robot collaboration: A user study. In *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pages 360–363, 2023.

[18] D. Coleman, I. Sucan, S. Chitta, and N. Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *arXiv preprint arXiv:1404.3785*, 2014.

[19] E. Coronado, T. Kiyokawa, G. A. G. Ricardez, I. G. Ramirez-Alpizar, G. Venture, and N. Yamanobe. Evaluating quality in human-robot interaction: A systematic search and classification of performance and human-centered factors, measures and metrics towards an industry 5.0. *Journal of Manufacturing Systems*, 63:392–410, 2022.

[20] L. S. Dalenogare, G. B. Benitez, N. F. Ayala, and A. G. Frank. The expected contribution of industry 4.0 technologies for industrial performance. *International Journal of production economics*, 204:383–394, 2018.

[21] J. DelPreto, J. I. Lipton, L. Sanneman, A. J. Fay, C. Fourie, C. Choi, and D. Rus. Helping robots learn: a human-robot master-apprentice model using demonstrations via virtual reality teleoperation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10226–10233. IEEE, 2020.

[22] M. Dianatfar, J. Latokartano, and M. Lanz. Review on existing vr/ar solutions in human–robot collaboration. *Procedia CIRP*, 97:407–411, 2021.

[23] N. Dimitropoulos, T. Togias, N. Zacharaki, G. Michalos, and S. Makris. Seamless human–robot collaborative assembly using artificial intelligence and wearable devices. *Applied Sciences*, 11(12):5699, 2021.

[24] X. Dongye, H. Guo, Y. Bao, and D. Weng. Gaussian replacement: Gaussians-mesh joint rendering for real-time vr interaction. In Y. Wang and H. Huang, editors, *Image and Graphics Technologies and Applications*, pages 312–326, Singapore, 2025. Springer Nature Singapore.

[25] J. S. Dyrstad, E. R. Øye, A. Stahl, and J. R. Mathiassen. Teaching a robot to grasp real fish by imitation learning from a human supervisor in virtual reality. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7185–7192. IEEE, 2018.

[26] H. Fang, S. Ong, and A. Nee. Interactive robot trajectory planning and simulation using augmented reality. *Robotics and Computer-Integrated Manufacturing*, 28(2):227–237, 2012.

[27] M. Farajtabar and M. Charbonneau. The path towards contact-based physical human–robot interaction. *Robotics and Autonomous Systems*, page 104829, 2024.

[28] Y. Feddoul, N. Ragot, F. Duval, V. Havard, D. Baudry, and A. Assila. Exploring human-machine collaboration in industry: A systematic literature review of digital twin and robotics interfaced with extended reality technologies. *The International Journal of Advanced Manufacturing Technology*, 129(5):1917–1932, 2023.

[29] A. Ferrari and K. Willcox. Digital twins in mechanical and aerospace engineering. *Nature Computational Science*, 4(3):178–183, 2024.

[30] J. Gao, C. Gu, Y. Lin, Z. Li, H. Zhu, X. Cao, L. Zhang, and Y. Yao. Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing. In *European Conference on Computer Vision*, pages 73–89. Springer, 2024.

[31] N. Gavish, T. Gutiérrez, S. Webel, J. Rodríguez, M. Peveri, U. Bockholt, and F. Tecchia. Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks. *Interactive Learning Environments*, 23(6):778–798, 2015.

[32] M. Ghobakhloo, H. A. Mahdiraji, M. Iranmanesh, and V. Jafari-Sadeghi. From industry 4.0 digital manufacturing to industry 5.0 digital society: A roadmap toward human-centric, sustainable, and resilient production. *Information Systems Frontiers*, pages 1–33, 2024.

[33] M. A. Goodrich, A. C. Schultz, et al. Human–robot interaction: a survey. *Foundations and trends® in human–computer interaction*, 1(3):203–275, 2008.

[34] M. Gu, E. Croft, and A. Cosgun. Ar point &click: An interface for setting robot navigation goals. In *International Conference on Social Robotics*, pages 38–49. Springer, 2022.

[35] A. Guédon and V. Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024.

[36] J. Guo, J. Leng, J. L. Zhao, X. Zhou, Y. Yuan, Y. Lu, D. Mourtzis, Q. Qi, S. Huang, X. Song, et al. Industrial metaverse towards industry 5.0: Connotation, architecture, enablers, and challenges. *Journal of Manufacturing Systems*, 76:25–42, 2024.

[37] K. C. Hoang, W. P. Chan, S. Lay, A. Cosgun, and E. Croft. Virtual barriers in augmented reality for safe and effective human-robot cooperation in manufacturing. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1174–1180, 2022.

[38] J. Hong, Z. Zhang, A. M. S. Enayati, and H. Najjaran. Human-robot skill transfer with enhanced compliance via dynamic movement primitives, 2023.

[39] C. Hsu, Y.-C. Sun, K.-Y. Lee, and C.-Y. Huang. Will neural 3d object representations be the silver bullet for improving vr experience in hmds? In *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 231–234. IEEE, 2024.

[40] L. Huang, M. Kapteyn, and K. E. Willcox. Digital twin: Graph formulations for managing complexity and uncertainty. In *2024 IEEE Smart World Congress (SWC)*, pages 2141–2148. IEEE, 2024.

[41] Z. Huang, Y. Shen, J. Li, M. Fey, and C. Brecher. A survey on ai-driven digital twins in industry 4.0: Smart manufacturing and advanced robotics. *Sensors*, 21(19), 2021.

[42] T. Inamura and Y. Mizuchi. Sigverse: A cloud-based vr platform for research on multimodal human-robot interaction. *Frontiers in Robotics and AI*, 8:549360, 2021.

[43] M. Z. Iqbal, E. Mangina, and A. G. Campbell. Exploring the real-time touchless hand interaction and intelligent agents in augmented reality learning applications. In *2021 7th International Conference of the Immersive Learning Research Network (iLRN)*, pages 1–8. IEEE, 2021.

[44] A. Jackson, B. D. Northcutt, and G. Sukthankar. The benefits of immersive demonstrations for teaching robots. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 326–334. IEEE, 2019.

[45] R. Jahanmahin, S. Masoud, J. Rickli, and A. Djuric. Human-robot interactions in manufacturing: A survey of human behavior modeling. *Robotics and Computer-Integrated Manufacturing*, 78:102404, 2022.

[46] Y. Jiang, S. Yin, K. Li, H. Luo, and O. Kaynak. Industrial applications of digital twins. *Philosophical Transactions of the Royal Society A*, 379(2207):20200360, 2021.

[47] Y. Jiang, C. Yu, T. Xie, X. Li, Y. Feng, H. Wang, M. Li, H. Lau, F. Gao, Y. Yang, and C. Jiang. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery.

[48] Y. Karpichev, T. Charter, J. Hong, A. M. Soufi Enayati, H. Honari, M. G. Tamizi, and H. Najjaran. Extended reality for enhanced human-robot collaboration: a human-in-the-loop approach. In *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, pages 1991–1998, 2024.

[49] Y. Karpichev, M. C. Zaouali, T. Charter, et al. A deployable and scalable ros-docker framework for multi-platform digital twin applications. TechRxiv, April 2025. Preprint.

[50] F. Kennel-Maushart, R. Poranne, and S. Coros. Multi-arm payload manipulation via mixed reality. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11251–11257. IEEE, 2022.

[51] F. Kennel-Maushart, R. Poranne, and S. Coros. Interacting with multi-robot systems via mixed reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11633–11639. IEEE, 2023.

[52] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[53] B. Kerbl, A. Meuleman, G. Kopanas, M. Wimmer, A. Lanvin, and G. Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics*, 43(4), July 2024.

[54] C. Kleinbeck, H. Schieber, K. Engel, R. Gutjahr, and D. Roth. Multi-layer gaussian splatting for immersive anatomy visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2025.

[55] C. Kleinbeck, H. Zhang, B. D. Killeen, D. Roth, and M. Unberath. Neural digital twins: reconstructing complex medical environments for spatial planning in virtual reality. *International Journal of Computer Assisted Radiology and Surgery*, 19(7):1301–1312, 2024.

[56] V. Kuts, J. A. Marvel, M. Aksu, S. L. Pizzagalli, M. Sarkans, Y. Bondarenko, and T. Otto. Digital twin as industrial robots manipulation validation tool. *Robotics*, 11(5):113, 2022.

[57] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann. Industry 4.0. *Business & information systems engineering*, 6:239–242, 2014.

[58] M. E. Latoschik, D. Roth, D. Gall, J. Achenbach, T. Waltemate, and M. Botsch. The effect of avatar realism in immersive social virtual realities. In *Proceedings of the 23rd ACM symposium on virtual reality software and technology*, pages 1–10, 2017.

[59] Y. J. Lee and Y. G. Ji. Effects of visual realism on avatar perception in immersive and non-immersive virtual environments. *International Journal of Human–Computer Interaction*, 41(7):4362–4375, 2025.

[60] K. Li, D. Chappell, and N. Rojas. Immersive demonstrations are the key to imitation learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5071–5077, 2023.

[61] X. Li, J. Li, Z. Zhang, R. Zhang, F. Jia, T. Wang, H. Fan, K.-K. Tseng, and R. Wang. Robogsim: A real2sim2real robotic gaussian splatting simulator, 2024.

[62] Y. Liu, S. Ong, and A. Nee. State-of-the-art survey on digital twin implementations. *Advances in Manufacturing*, 10(1):1–23, 2022.

[63] M. B. Luebbers, C. Brooks, C. L. Mueller, D. Szafir, and B. Hayes. Arc-lfd: Using augmented reality for interactive long-term robot skill maintenance via constrained learning from demonstration. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3794–3800. IEEE, 2021.

[64] H. Mao, Z. Xu, S. Wei, Y. Quan, N. Deng, and X. Yang. Live-gs: Llm powers interactive vr by enhancing gaussian splatting, 2024.

[65] A. Martínez-Gutiérrez, J. Díez-González, H. Perez, and M. Araújo. Towards industry 5.0 through metaverse. *Robotics and Computer-Integrated Manufacturing*, 89:102764, 2024.

[66] P. Melo, R. Arrais, and G. Veiga. Development and deployment of complex robotic applications using containerized infrastructures. In *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, pages 1–8, 2021.

[67] S. Mihai, M. Yaqoob, D. V. Hung, W. Davis, P. Towakel, M. Raza, M. Karamanoglu, B. Barn, D. Shetve, R. V. Prasad, H. Venkataraman, R. Trestian, and H. X. Nguyen. Digital twins: A survey on enabling technologies, challenges, trends and future prospects. *IEEE Communications Surveys & Tutorials*, 24(4):2255–2291, 2022.

[68] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[69] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies*, volume 2351, pages 282–292. Spie, 1995.

[70] N. Moenne-Loccoz, A. Mirzaei, O. Perel, R. de Lutio, J. Martinez Esturo, G. State, S. Fidler, N. Sharp, and Z. Gojcic. 3d gaussian ray tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics (TOG)*, 43(6):1–19, 2024.

[71] D. Mukherjee, K. Gupta, L. H. Chang, and H. Najjaran. A survey of robot learning strategies for human-robot collaboration in industrial settings. *Robotics and Computer-Integrated Manufacturing*, 73:102231, 2022.

[72] D. Mukherjee, K. Gupta, and H. Najjaran. An ai-powered hierarchical communication framework for robust human-robot collaboration in industrial settings. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1321–1326. IEEE, 2022.

[73] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[74] H. Nemlekar, N. Dhanaraj, A. Guan, S. K. Gupta, and S. Nikolaidis. Transfer learning of human preferences for proactive robot assistance in assembly tasks. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pages 575–583, 2023.

[75] M. Newman, B. Gatersleben, K. Wyles, and E. Ratcliffe. The use of virtual reality in environment experiences and the importance of realism. *Journal of Environmental Psychology*, 79:101733, 2022.

[76] J. Nielsen. Ten usability heuristics. 2005.

[77] S. Papanastasiou, N. Kousi, P. Karagiannis, C. Gkournelos, A. Papavasileiou, K. Dimoulas, K. Baris, S. Koukas, G. Michalos, and S. Makris. Towards seamless human robot collaboration: integrating multimodal interaction. *The International Journal of Advanced Manufacturing Technology*, 105:3881–3897, 2019.

[78] S. Patil, V. Vasu, and K. Srinadh. Advances and perspectives in collaborative robotics: a review of key technologies and emerging trends. *Discover Mechanical Engineering*, 2(1):13, 2023.

[79] B. Perroud, S. Régnier, A. Kemeny, and F. Mérienne. Model of realism score for immersive vr systems. *Transportation research part F: Traffic psychology and behaviour*, 61:238–251, 2019.

[80] PickNik Robotics. Concepts - moveit documentation, 2024. Accessed: 2025-04-10.

[81] D. G. Pivoto, L. F. De Almeida, R. da Rosa Righi, J. J. Rodrigues, A. B. Lugli, and A. M. Alberti. Cyber-physical systems architectures for industrial internet of things applications in industry 4.0: A literature review. *Journal of manufacturing systems*, 58:176–192, 2021.

[82] S. Pizzagalli, V. Kuts, and T. Otto. User-centered design for human-robot collaboration systems. In *IOP Conference Series: Materials Science and Engineering*, page 012011. IOP Publishing, 2021.

[83] M. Pizzo, E. Viola, F. Solari, and M. Chessa. Evaluation of 3d reconstruction techniques for the blending of real and virtual environments. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 360–367. IEEE, 2024.

[84] A. M. Potdar, N. D G, S. Kengond, and M. M. Mulla. Performance evaluation of docker container and virtual machine. *Procedia Computer Science*, 171:1419–1428, 2020. Third International Conference on Computing and Network Communications (CoCoNet'19).

[85] A. Pranckevičius. Unity gaussian splatting, 2024.

[86] S. Qiu, B. Xie, Q. Liu, and P.-A. Heng. Creating virtual environments with 3d gaussian splatting: A comparative study, 2025.

[87] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[88] C. P. Quintero, S. Li, M. K. Pan, W. P. Chan, H. M. Van der Loos, and E. Croft. Robot programming through augmented trajectories in augmented reality. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1838–1844. IEEE, 2018.

[89] S. M. Ragil, Z. Tieling, and S. Kiridena. Overview of ergonomics and safety aspects of human-cobot interaction in the manufacturing industry. In *International Conference on Informatics, Technology and Engineering*, volume 21, page 401, 2023.

[90] A. Rega, C. Di Marino, A. Pasquariello, F. Vitolo, S. Patalano, A. Zanella, and A. Lanzotti. Collaborative workplace design: A knowledge-based approach to promote human–robot collaboration and multi-objective layout optimization. *Applied Sciences*, 11(24):12147, 2021.

[91] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[92] J. Saukkoriipi, T. Heikkilä, J. M. Ahola, T. Seppälä, and P. Isto. Programming and control for skill-based robots. *Open Engineering*, 10(1):368–376, 2020.

[93] E. Senft, M. Hagenow, K. Welsh, R. Radwin, M. Zinn, M. Gleicher, and B. Mutlu. Task-level authoring for remote robot teleoperation. *Frontiers in Robotics and AI*, 8:707149, 2021.

[94] A. Sharma and B. J. Singh. Evolution of industrial revolutions: A review. *International Journal of Innovative Technology and Exploring Engineering*, 9(11):66–73, 2020.

[95] A. Shojaeinasab, T. Charter, M. Jalayer, M. Khadivi, O. Ogunfowora, N. Raiyani, M. Yaghoubi, and H. Najjaran. Intelligent manufacturing execution systems: A systematic review. *Journal of Manufacturing Systems*, 62:503–522, 2022.

[96] D. Silva, J. Rafael, and A. Fonte. Toward optimal virtualization: An updated comparative analysis of docker and lxd container technologies. *Computers*, 13(4):94, 2024.

[97] J. Smith and R. Nair. The architecture of virtual machines. *Computer*, 38(5):32–38, 2005.

[98] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, July 2006.

[99] J. E. Solanes, A. Muñoz, L. Gracia, A. Martí, V. Girbés-Juan, and J. Tornero. Teleoperation of industrial robot manipulators based on augmented reality. *The International Journal of Advanced Manufacturing Technology*, 111:1077–1097, 2020.

[100] A. D. Souchet, D. Lourdeaux, A. Pagani, and L. Rebenitsch. A narrative review of immersive virtual reality's ergonomics and risks at the workplace: cybersickness, visual fatigue, muscular fatigue, acute stress, and mental overload. *Virtual Reality*, 27(1):19–50, 2023.

[101] M. Speicher, B. D. Hall, and M. Nebeling. What is mixed reality? In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–15, 2019.

[102] D. Sun, A. Kiselev, Q. Liao, T. Stoyanov, and A. Loutfi. A new mixed-reality-based teleoperation system for telepresence and maneuverability enhancement. *IEEE Transactions on Human-Machine Systems*, 50(1):55–67, 2020.

[103] 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 *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–33, 2022.

[104] K. A. Szczurek, R. Cittadini, R. M. Prades, E. Matheson, and M. Di Castro. Enhanced human–robot interface with operator physiological parameters monitoring and 3d mixed reality. *IEEE Access*, 11:39555–39576, 2023.

[105] 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. *Engineering*, 5(4):653–661, 2019.

[106] G. Tsamis, G. Chantziaras, D. Giakoumis, I. Kostavelis, A. Kargakos, A. Tsakiris, and D. Tzovaras. Intuitive and safe interaction in multi-user human robot collaboration environments through augmented reality displays. In *2021 30th IEEE international conference on robot & human interactive communication (RO-MAN)*, pages 520–526. IEEE, 2021.

[107] D. Ungureanu, F. Bogo, S. Galliani, P. Sama, X. Duan, C. Meekhof, J. Stühmer, T. J. Cashman, B. Tekin, J. L. Schönberger, P. Olszta, and M. Pollefeys. Hololens 2 research mode as a tool for computer vision research, 2020.

[108] R. van Dinter, B. Tekinerdogan, and C. Catal. Predictive maintenance using digital twins: A systematic literature review. *Information and Software Technology*, 151:107008, 2022.

[109] S. Vi, T. S. da Silva, and F. Maurer. User experience guidelines for designing hmd extended reality applications. In *Human-Computer Interaction–INTERACT 2019: 17th IFIP TC 13 International Conference, Paphos, Cyprus, September 2–6, 2019, Proceedings, Part IV 17*, pages 319–341. Springer, 2019.

[110] J. Waczyńska, P. Borycki, S. Tadeja, J. Tabor, and P. Spurek. Games: Mesh-based adapting and modification of gaussian splatting. *arXiv preprint arXiv:2402.01459*, 2024.

[111] D. Weber, W. Fuhl, E. Kasneci, and A. Zell. Multiperspective teaching of unknown objects via shared-gaze-based multimodal human-robot interaction.

In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pages 544–553, 2023.

[112] D. Wei, B. Huang, and Q. Li. Multi-view merging for robot teleoperation with virtual reality. *IEEE Robotics and Automation Letters*, 6(4):8537–8544, 2021.

[113] R. White and H. Christensen. *ROS and Docker*, pages 285–307. Springer International Publishing, Cham, 2017.

[114] K. Willcox and B. Segundo. The role of computational science in digital twins. *Nature Computational Science*, 4(3):147–149, 2024.

[115] Y. Wolf, A. Bracha, and R. Kimmel. Surface reconstruction from gaussian splatting via novel stereo views. *arXiv e-prints*, pages arXiv–2404, 2024.

[116] T. Xie, Z. Zong, Y. Qiu, X. Li, Y. Feng, Y. Yang, and C. Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024.

[117] M. Xu, J. M. David, S. H. Kim, et al. The fourth industrial revolution: Opportunities and challenges. *International journal of financial research*, 9(2):90–95, 2018.

[118] X. Yan, Y. Jiang, C. Chen, L. Gong, M. Ge, T. Zhang, and X. Li. A complementary framework for human–robot collaboration with a mixed ar–haptic interface. *IEEE Transactions on Control Systems Technology*, 2023.

[119] J. Yang, Y. Liu, and P. L. Morgan. Human-machine interaction towards industry 5.0: Human-centric smart manufacturing. *Digital Engineering*, page 100013, 2024.

[120] M. H. Zafar, E. F. Langås, and F. Sanfilippo. Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry 5.0 for smart manufacturing: A state-of-the-art review. *Robotics and Computer-Integrated Manufacturing*, 89:102769, 2024.

[121] M. C. Zaouali, T. Charter, Y. Karpichev, B. Haworth, and H. Najjjaran. A study of the framework and real-world applications of language embedding for 3d scene understanding. *arXiv*, 2025.

[122] M. K. Zein, M. Al Aawar, D. Asmar, and I. H. Elhajj. Deep learning and mixed reality to autocomplete teleoperation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4523–4529. IEEE, 2021.

[123] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635. IEEE, 2018.

[124] Y. Zhang, B. Orthmann, M. C. Welle, J. Van Haastregt, and D. Kragic. Llm-driven augmented reality puppeteer: Controller-free voice-commanded robot teleoperation. In A. Coman and S. Vasilache, editors, *Social Computing and Social Media*, pages 97–112, Cham, 2025. Springer Nature Switzerland.

[125] Y. Zheng, X. Chen, Y. Zheng, S. Gu, R. Yang, B. Jin, P. Li, C. Zhong, Z. Wang, L. Liu, et al. Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping. *IEEE Robotics and Automation Letters*, 2024.

[126] T. Zhou and J. P. Wachs. Early turn-taking prediction with spiking neural networks for human robot collaboration. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3250–3256. IEEE, 2018.

[127] M. Zimmerman, S. Bagchi, J. Marvel, and V. Nguyen. An analysis of metrics and methods in research from human-robot interaction conferences, 2015–2021. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 644–648. IEEE, 2022.