Robot Operation by Ambient Meta World

青木, 惇季

Robot Operation by Ambient Meta World

Junki Aoki

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Department of Information Science and Technology,
Graduate School of Information Science and Electrical Engineering,
Kyushu University

July 2024

Abstract

This dissertation proposes *Ambient Meta World (AMW)* to enhance the usability of robotic systems by integrating virtual spaces within real-world operations in digital twin (DT).

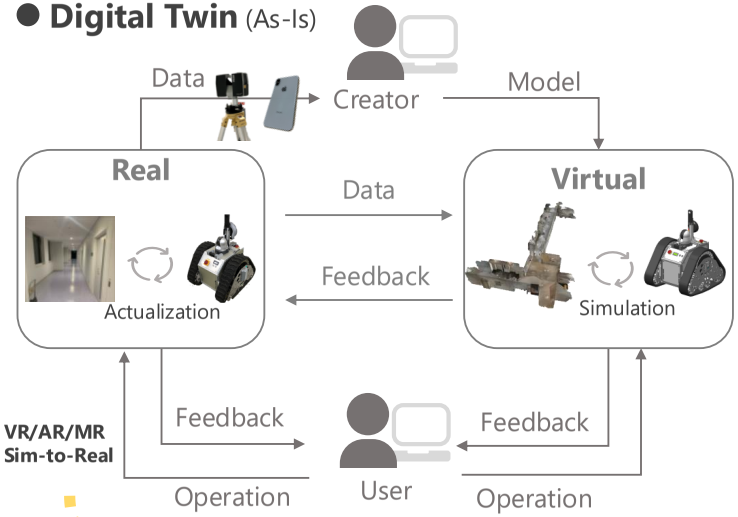
Traditional robotic systems utilizing DT and Cyber-Physical Systems can simulate robot actions in cyberspace based on information gathered from the physical environment and then feed the results back into the physical space. This setup is helpful as it allows for robot operation simulations without the constraints of the physical world.

However, the DT system requires users to engage in complex operations and skills. Users need to be aware of both cyberspace and physical space, understanding and making decisions based on the robot's status across these dimensions. This dissertation proposes reevaluating the paths of user interaction that require awareness of both cyberspace and physical space, aiming to enable users to enjoy the benefits of the digital twin system through interactions solely with the physical space.

To seamlessly integrate virtual spaces without user awareness, this dissertation proposes a two-pronged approach: *Illusional Reality*, which is a system feedback to humans, and *Sim-in-Real*, which is a model construction automation. Illusional Reality involves embedding virtual spaces within real spaces, making transitions seamless and seemingly manipulating only the physical space, while simulations occur in cyberspace. In sim-in-real, building cyberspace should be implicit, allowing immediate use without explicit setup. This dissertation aims to realize AMW by ensuring that users can engage with digital twins without perceiving a distinction between real and virtual spaces or undergoing cumbersome setup processes.

Thus, this effort explains the implementation of AMW in robotic systems and discusses its contribution to enhancing user convenience for those systems.

● Digital Twin (As-Is)

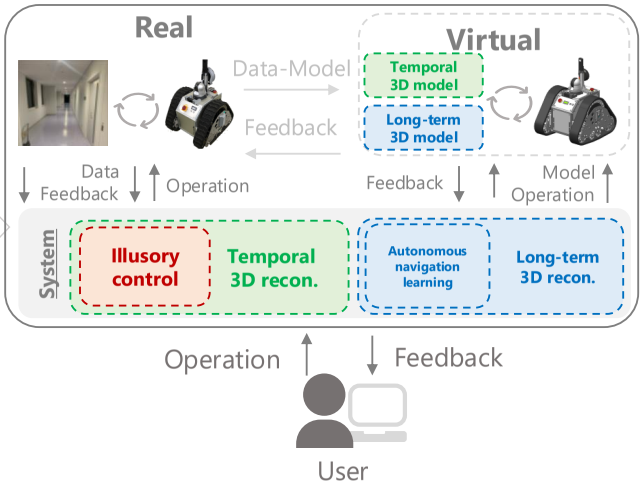


Background
- Advancements in 3D simulation
  - Creating a photorealistic model easily
- Expansion of
  1. target user
  2. applicable area

Aim
- Improvement of usability
  1. w/o expertise
  2. w/o efforts

Task
- Realization of ***Latent Digital Twin***
  (Technologies merging into daily life)
  A. Seamless integration of R/V feedback
  B. Automation of data-model process

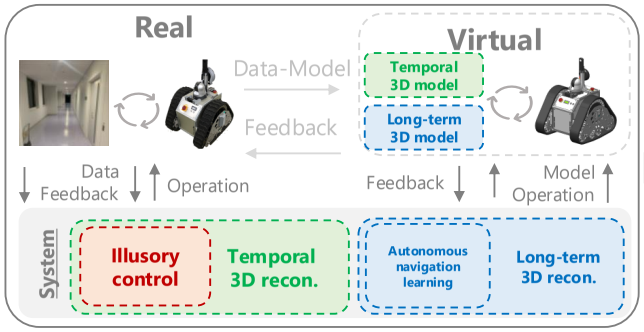
● Ambient Meta World (To-Be)



● Contribution

Illusional Reality (for A. Feedback)
Seamless transition between R/V with advantage in V
- **Improving operation usability in obstacle avoidance**

Sim-in-Real (for B. Data-Model)
Sim. creation based on real sensor obtained during operations
- One-shot image → **Temporal 3D model** → **Integration w/ IR**
- One-shot image sequence → Long-term 3D model → **Auto. Nav.**



● Future Work

Integration with Generative Models
- Improvement of reality of environment
  (Illusion for human and agents)
- Generation of dynamic objects and integration with virtual

Acknowledgements

This dissertation is a doctoral thesis on the technology developed in a joint research project between the Kurazume laboratory of Kyushu University and Ricoh Co., Ltd, and was written based on [1, 2, 3, 4]. This achievement would not have been possible without the support of everyone who has associated with me.

First, I would like to thank Professor Ryo Kurazume for his tremendous support of my research. He respected what we at Ricoh wanted to do as the direction of technological development, and he provided us with a very exciting idea of Illusory Control. He gave us appropriate advice in our daily research discussion and supported our development. Under his guidance, advancing our research daily has significantly boosted my confidence as a researcher. Professor Atsushi Shimada was a chief examiner of my doctoral dissertation, and Professor Kazuo Kiguchi was a sub examiner of my doctoral dissertation. I would also like to thank them for taking the time to review my dissertation and for providing constructive advice on the future direction of my research. I received a lot of advice from Assistant Professor Kohei Matsumoto on how to implement modules of the Robot Operation System and on the application of reinforcement learning in the development of autonomous functions. I would also like to thank Associate Professor Qi An, Associate Professor Akihiro Kawamura, and Assistant Professor Kazuto Nakashima for their constructive advice on my research from the perspective of their own fields of expertise. The members of the Kurazume laboratory are highly motivated in their robotics research and shared exciting research updates during our weekly seminar time. Their motivation drove me to continue with my research and development. I also appreciate their cooperation in the user study. In addition, this dissertation is partially supported by JSPS KAKENHI Grant Numbers JP21K18701 and JP20H00230.

The members of Ricoh Co., Ltd. have supported and assisted me with my research and development. Mr. Yoshiaki Umetsu and Mr. Shinya Iguchi approved my admis-

sion to the doctoral program and encouraged my research and development achievements. I would like to thank Group Leader Ryota Yamashina for all the support he has given me, from my work at the Robotics Group to my daily life. I am very grateful to him for creating a work environment where I can focus on my research and development while keeping the project of robotics. Mr. Atsuo Kawaguchi approved my dispatch to Kyushu University when he was the director of the research institute and encouraged me to enter the doctoral program for working professionals. Mr. Fumihiro Sasaki of the same team consulted with me on the direction of my research and development and gave me a lot of appropriate advice that significantly promoted my research, especially in the development of autonomous navigation. Mr. Wataru Hatanaka and I can proceed with research and development while encouraging each other as colleagues in the company and as working PhDs, and he gave me emotional support. Mr. Asuto Taniguchi gave me unique ideas when sharing my progress and advice on writing the paper. Mr. Hiroshi Shimura, Mr. Tetsuro Sasamoto, Mr. Taku Kitahara, and Mr. Susumu Minagi came to Kyushu University, and they provided me with a crawler robot for the experimental platform. I also thank Mr. Mototsugu Muroi and Mr. Kento Hosaka for their advice on system implementation for mobile robots. In addition, Mr. Eichi Koizumi created a robot model to commemorate my achievement, encouraging my research activities.

Finally, I would like to thank my family. My wife Kayo, father Osamu, mother Fumiko, brother Sota, and grandmother Eiko, thank you for watching over me as I entered the workforce doctoral program.

My research results have been made possible by the support of many people. I would like to thank them for their support, along with other people whose names I could not mention in this short text.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

1.1.1 Digital Twin Robot System

One of the fundamental strategies to realize *Society 5.0* is the ”advanced integration of cyberspace and physical space” [9]. The concept of digital twins (DT) embodies this strategy by replicating the physical space in cyberspace, thereby digitalizing data from various physical fields and aiming to solve some problems within the framework of the *Cyber-Physical System (CPS)*. The concept of the DT has been considered for application in various fields, including manufacturing, automotive, aerospace, residential & commercial, and retail & consumer goods companies. The market size of the DT is forecasted to reach 91.92 billion dollars by 2028, with the platform expected to be further utilized in the future, backed by the increase in remote work and the growing demand for labor-saving measures.

According to the ”Research and Development Trends in Digital Twins Domesti-

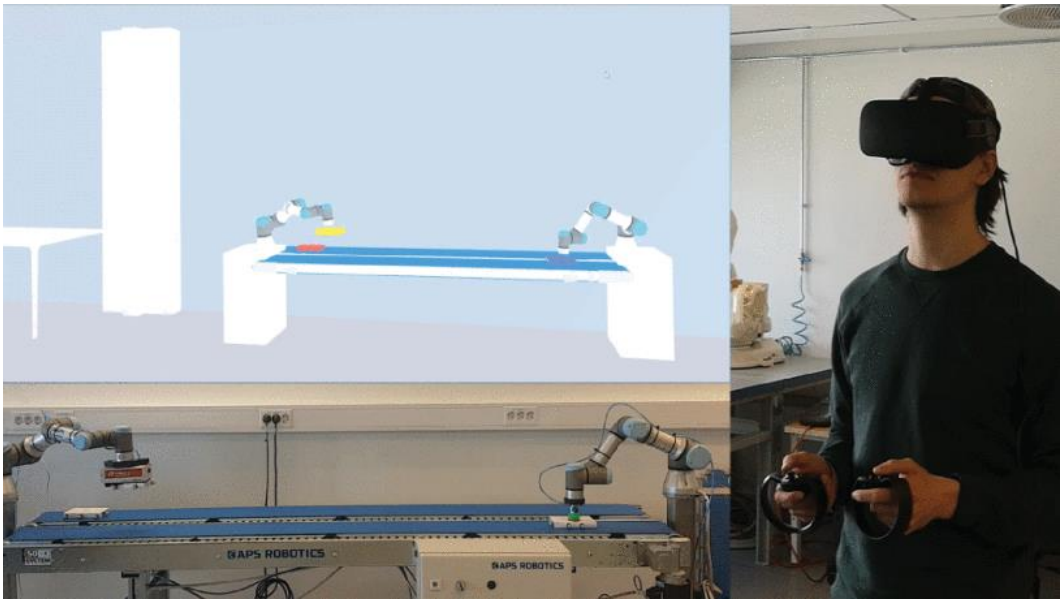


Figure 1.1: VR teleoperation via DT [5]

cally and Internationally" issued by JST-CRDS [10], DT comprises data, models, and interfaces. Data refers to the data obtained through advanced measurements and observations in physical spaces and data covering the entire lifecycle, which varies widely. Models refer to various computational and representational models corresponding to physical spaces, including physical, statistical, machine learning, geometric, and visualization models (such as 3D simulations and extended reality). Interfaces enable connections and interactions between the DT and applications. These components allow for the simulation and prediction in physical spaces within cyberspace, providing optimal solutions to physical spaces. Although the document defines various types of data, this research focuses on a DT system that is limited to visual-spatial information represented in 3D models, dealing specifically with color information obtained through devices such as cameras and geometric information derived from distance sensors, such as LiDAR, depth cameras and so forth. This dissertation refers to cyberspace as "virtual space" and physical space as "real space."

Research on robot operation systems leveraging DT is actively being conducted [11]. By replicating an exact virtual counterpart of the real space on a computer, it is possible to leverage the advantage of being free from physical constraints, allowing for low-risk activities such as visualizing future trajectories in Human-Robot Interac-

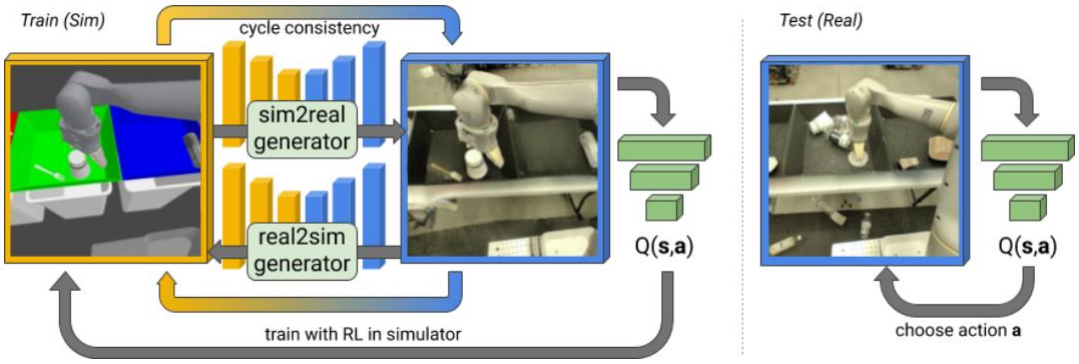


Figure 1.2: Example of sim-to-real architecture [6].

tion (HRI) for monitoring safety behavior and conducting simulations for autonomous robots navigation and manipulation.

In the context of teleoperation robots, the Virtual Reality (VR) environment allows for the presentation of action predictions, such as task instructions, which are widely used for directing complex tasks. Intended actions are simulated in a VR environment beforehand, and the results are then fed back to the actual machine [12, 5], as shown in Figure 1.1. Additionally, methods include using Augmented Reality (AR) and Mixed Reality (MR) for information presentation in the real space [13]. The ability to simulate robot actions in an environment without physical constraints results in benefits such as improved usability.

For autonomous robots, simulations of movements within a simulated space are possible. Learning-based autonomous navigation is a widely applicable means of movement, enabling simulations within virtual spaces to anticipate how environments will appear as the robot moves through them. By obtaining the necessary data for autonomous navigation from virtual spaces rather than real spaces, efficiencies in autonomous data collection are anticipated. Additionally, the policy for autonomous mobile robots learned through simulations in virtual space, is transferred to adapt to the real world using a method called simulation-to-real (sim-to-real) [6, 14], as shown in Figure 1.2.

Thus, robotics utilizing DT can be expected to serve as a platform for solutions to various societal problems, thanks to its data analysis and simulation capability without physical constraints. This approach not only enhances the efficiency and effectiveness of solving complex problems but also opens new avenues for innovation in addressing challenges that are critical to society's progress.

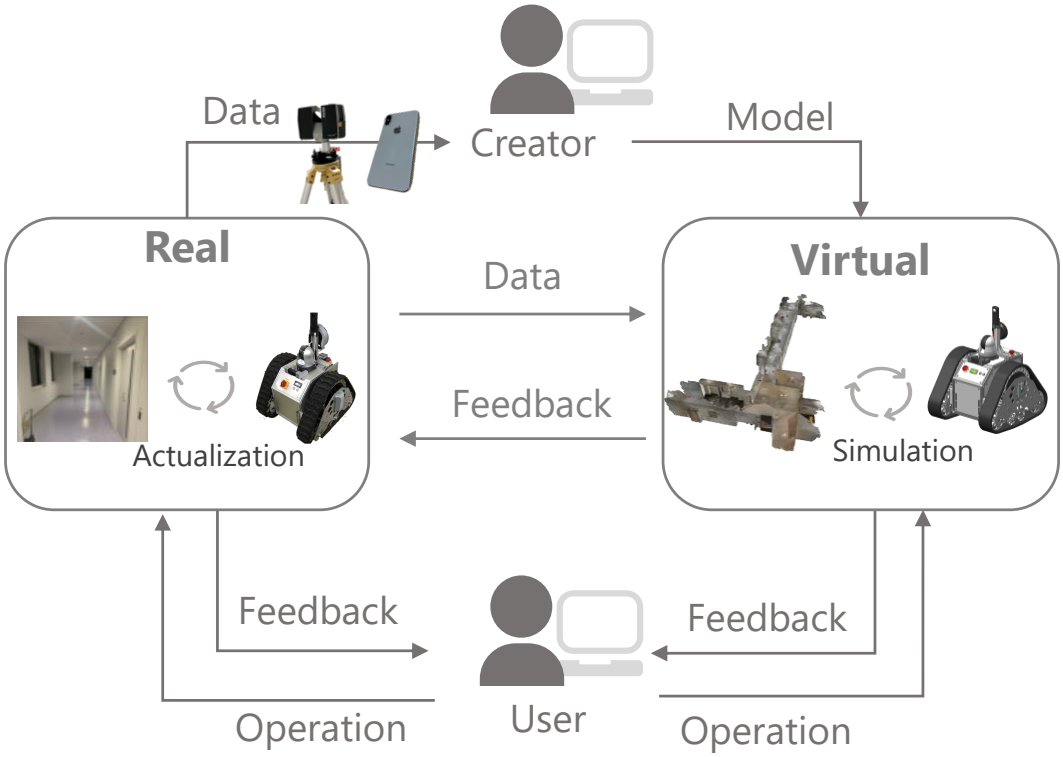


Figure 1.3: The data flow of the DT.

1.1.2 Issues

While robot operations utilizing DT are beneficial, their users face complexity issues related to user interfaces and the extensive preparation based on specialized and advanced knowledge.

This research examines system designs that allow DT systems, conventionally complex, to be realized through simple user interactions. It is desirable to provide interaction designs that can be used by people without specialized knowledge and that allow the use of DT systems without extensive preparation efforts. The end-users focused on in this dissertation are those who, through simple practice, can operate robots but do not possess the technical knowledge required for tasks such as creating 3D models of virtual spaces, constructing environmental maps for autonomous robot navigation, or receiving specialized education in robot operation.

The flow of a conventional robot system utilizing DT is illustrated in Figure 1.3. In realizing the simple interaction design, there are two problems with the conventional system flow of DT.

User interface: The first issue is the user burden in the feedback and operation processes in Figure 1.3, which we collectively define as the user interface problem. In robot operations via DT, it is common to use technologies such as VR, AR, and MR. In conventional robot operations via DT, it is assumed that the user is responsible for the robot's actions. For instance, in remote surgical robots, the VR space provides real-time feedback of the actual conditions in the physical space [15]. Consequently, users are affected by communication delays and must perform operations with this understanding. This places a significant burden on the user. Additionally, when there is a conflict of intentions between the user and the robot system, it is often unclear who guarantees the robot's actions, potentially leading to confusion. Similarly, AR and MR also generally assume real-time feedback of the actual conditions in the physical space, which can lead to the aforementioned issues.

Thus, in conventional robot operations, the user interface requires the user to understand the conditions of both the virtual and real spaces and to take responsibility for the robot's actions. This necessitates specialized operational skills and expertise focused on robot operation. Therefore, improvements that offer simpler operation than VR/MR/AR technologies are necessary.

Model construction: The second issue involves the specialized knowledge required to construct 3D models and the workload involved in preparation. As depicted in [16], there are typically two key roles of humans involving DT systems: the users who wish to solve problems using the system and creators who prepare everything beforehand.

Creators are needed to perform preparatory work to construct the 3D models, involving bringing devices equipped with sensors to collect color and distance information into the real space. After sensing, the data must be processed and incorporated into a computer for editing and creating the 3D models.

These tasks require specialized knowledge to create 3D models and significant work. Furthermore, an approach is needed for knowledge, such as reducing the visual domain gap and transferring from virtual to real spaces. Research in the field of sim-to-real, which aims to bridge the visual gap between virtual and real spaces, is well-established. Sim-to-real is a technology that operates under the assumption that both simulation and real environments are prepared. Typically, in sim-to-real, data collection is necessary in both the simulation and real environments, thus necessitating significant effort. Thus, the capability to construct DT systems has traditionally

been limited to those with specialized knowledge and the ability to collect data and adjust models.

Ideally, even end-users without specialized knowledge should be able to construct such DT systems, making the technology more accessible and user-friendly.

1.2 Related Work

1.2.1 3D Reconstruction

Employing 3D Reconstruction technology is a promising approach for constructing 3D models for DT systems. Humans use sensing devices to acquire data in real spaces and can efficiently construct 3D models based on the color and distance information observed or estimated during this process.

3D reconstruction technologies include image-based and laser-based methods. Image-based approaches, utilizing stereo cameras and Structure from Motion (SfM) [17], have evolved towards denser point cloud generation through multi-view Stereo (MVS) [18], improving reconstruction fidelity. However, image-based methods like MVS face challenges in replicating complex scenes. Laser-based methods, leveraging laser scanners like LiDAR, gather environmental distance information to generate point clouds, enabling high-precision geometric reproduction even in complex environments.

Recent advances in deep learning have enabled techniques that combine the strengths of both laser-based and image-based methods. Technologies like image style transfer have been developed, allowing for adjustments to the appearance of 3D models to make them resemble their real-world counterparts more closely. For instance, CycleGAN [19] learns to transform images between two domains, adapting the appearance to match the other domain [6, 20]. This is particularly useful in addressing the sim-to-real gap in autonomous movement simulations, where technology continually evolves to bridge the visual gap between virtual and real spaces.

The development of 3D reconstruction as a product has also been remarkable. In 2020, Apple Inc. released the iPhone 12 Pro equipped with LiDAR. This addition of distance-measuring capabilities to a device used daily by people, combined with RGB camera functionality, has made it possible for individuals to easily create 3D models.

In 2020, the Neural Radiance Fields (NeRF) [21] technology was introduced, en-

Figure 1.4: A rendering result of the large scale reconstruction [7].

abling the reconstruction of photorealistic 3D models from images captured with RGB cameras. Some enhancements by various researchers, such as a large-scale 3D reconstruction, are shown in Figure 1.4, have led to lighter models and faster learning convergence, and NeRFs have the potential to handle complex scenes effectively [7]. NeRF technology has become accessible not only to engineers but also to general users through applications like Luma AI and nerfstudio, allowing anyone to start NeRF training with just a smartphone to capture environmental or object images, facilitating easy creation of 3D models.

Previously, it was assumed that creators of DT, equipped with specialized skills and dedicated effort, were necessary for their construction [16]. Under this premise, DT was primarily developed for structured environments like factories and construction sites where critical information is concentrated. However, this premise made it difficult to create DT easily in all types of environments.

However, in recent years, technologies such as NeRF and applications such as LumaAI and nerfstudio have emerged, making it possible for consumers without specialized knowledge to create 3D models easily. Consequently, systems applying the DT concept might evolve to allow end-users without specialized skills to create them without much burden. This means that end-users could take on the role of creators. As a result, DT, previously limited to specific environments, is likely to be developed for various types of users in various spaces.

1.2.2 Human-Robot Interaction

In DT systems, a user interface is required to provide users with feedback on the status of the virtual and real spaces. Technologies used for the user interface in such systems include Extended Reality (xR) technologies such as VR [22], AR, and MR [23]. VR

immerses users in a virtual space through computer-connected displays, and the robot's state and future intentions are fed back to the user in virtual spaces. In contrast, AR and MR overlay virtual objects onto the real world, and the robot's state and future intentions are fed back to the user in real spaces.

The concept of substitutional reality (SR) has also been proposed [24]. SR is a concept that does not require users to recognize whether something is real or virtual but rather feeds back images of past experiences as if they were happening here and now; and basic verification of this concept is underway, SR can create feedback by adding and subtracting information to and from information in the real world. In summary, when illustrating the positioning of VR/AR/MR/SR, it would be represented as shown in Figure 1.5.

SR is primarily seen as promising for entertainment and healthcare purposes, yet it embodies a crucial concept beneficial to the development of DT. SR blurs the line between virtual and real spaces, allowing users to interact with the system as naturally as they would with the real spaces without needing to discern whether what they are viewing is real or virtual. This seamless integration would enhance the usability of CPS utilizing DT by making interactions more intuitive and integrated into real-world contexts.

When Mark Weiser advocated for ubiquitous computing, they said that the most profound technologies are those that disappear, and they weave themselves into the fabric of everyday life until they are indistinguishable from it [25]. The widespread adoption of smartphones around 2010, integrating information touchpoints into devices already used daily, such as the telephone, exemplifies this vision, seamlessly merging with everyday life. Additionally, Ambient Intelligence [26] is a concept where sensors installed in the environment detect the user's situation, and the system adjusts its behavior according to the user's conditions and preferences, aiming to enhance the user experience. Devices are seamlessly integrated into daily life, becoming invisible to the user, and the system automatically adjusts its behavior based on the user's situation. As a result, users can utilize the system without being particularly conscious of its usability. As mentioned in Section 1.2.1, as the cost of creating 3D models decreases, the number of users who can take advantage of the system will be expanded. However, to enable users to easily use DT, even without specialized knowledge or complex operability, it is essential to design familiar user interactions that adapt to the expanding user base.

DT systems are realized based on a virtual space that has yet to be seamlessly integrated into daily life and is consciously recognized under concepts like VR/AR/MR. While VR/AR/MR-based techniques are becoming increasingly prevalent, there is significant research potential in exploring how virtual spaces can smoothly blend into real-world contexts, offering value and enhancing system usability. This direction of research could effectively return the benefits of DT to society seamlessly.

1.2.3 Sim-to-Real

Especially in the context of autonomous robots learning their behaviors, bridging the visual gap between simulation environments and real spaces is a critical technology for improving the accuracy of robot operations, and it is studied within the field of Sim-to-Real. After a robot learns behaviors and generates an inference model in a simulation environment, directly applying this model in real space to perform operations can be challenging. It becomes necessary to shift the domain from data observed in the virtual space to data observed in the real space. Conventional methods for this issue have involved domain transformation techniques [6, 20, 27, 28], randomization [29], and adaptation methods [30]. All these methods require the independent creation of a virtual space in advance, separate from the real space. Furthermore, it is necessary to collect data within both the real and virtual spaces and learn models for performing domain shifts. Such processes demand specialized knowledge and significant effort in 3D model creation to create a virtual space that resembles a real space and data collection.

Ideally, these specialized skills and efforts would become unnecessary. Given the advancements in 3D reconstruction technologies, which now allow for easy utilization and high-quality rendering, designing robotic operation systems that maximize these benefits presents a valuable direction, making sophisticated robot operations accessible to a broader range of users without requiring them to have technical expertise and workloads.

1.3 Research Aim

The aim is to enhance the usability of robot operations by designing a system that sublimates the awareness of virtual space in the DT systems. This approach to sublimating awareness of virtual space is referred to as the *Ambient Meta World (AMW)*. This

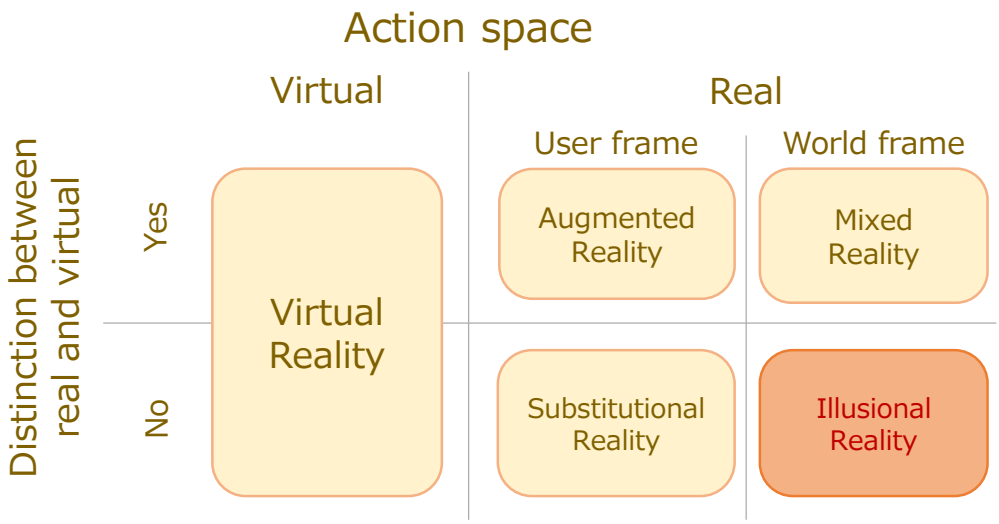


Figure 1.5: The positioning of xR.

research proposes the following two approaches to solve the aforementioned ”user interface” and ”model construction” problems, respectively. The approaches are as follows.

1.3.1 Illusional Reality

Regarding the user interface, unlike the VR/AR/MR discussed in Section 1.2.2, we consider a system design that utilizes the benefits of simulations in virtual space without the need for conscious awareness of it, similar to SR. One advantage of simulations in virtual space is the ability to test operations without physical constraints. In the real space, where physical constraints apply, it is challenging for users to operate robots smartly while being aware of the real space, as they must navigate obstacles and complex environments accurately. However, using simulations, users can easily instruct the expected operations. Ideally, if this could be presented in a way that the user is unaware of, as if they are operating within the real space, and the user can entrust the responsibility for the robot's actual actions to the system, it would be possible to provide a user-friendly system that balances the recognition of both virtual and real spaces and handles complex operations. By seamlessly presenting information that is convenient for the user without distinct boundaries between reality and simulation, we name this approach *Illusional Reality (IR)*. In this research, we specifically discuss the techniques for robotic operations under this concept, named *Illusory Control (IC)*.

In relation to the xR technologies discussed in Section 1.2.2, IR is positioned as shown in Figure 1.5. The user's activity space is based on the real space, and IR is characterized by presenting information in a way that does not make the boundary between the virtual and real spaces apparent. Figure 1.5 mainly summarizes the positioning from the perspective of how information is presented, but IR is also explained in terms of what information is presented, as shown in Figure 1.6. SR presents past images of a specific location. Previewed Reality (PR) [31] presents future information about the robot from the user's perspective, aiming to ensure safe coexistence between robots and users by showing the robot's future trajectory. IR, similar in nature to the information presented by PR, also presents future events. While PR presents the robot's future from the user's perspective, IR is characterized by presenting the future that the user expects from the robot's perspective.

Note that IR is not posited as a superior alternative to interfaces such as VR, AR, or MR; rather, IR is presented as one of several viable options within interface design. The appropriateness of VR, AR, MR, or IR depends on the user's level of expertise and personal preferences regarding operations. Consequently, this research does not undertake a comparative analysis of the performance across VR, AR, MR, and IR within the realm of robotic operations.

1.3.2 Sim-in-Real

Regarding the model construction process, it is desirable to automatically create 3D models based on data easily obtained in real space. Furthermore, embedding the process of creating 3D models into some steps of robot operation without consciously focusing on model creation can achieve a system design that reduces awareness of virtual space. Naming this integration as *Sim-in-Real,* this research explains two examples of applying Sim-in-Real to robotic operation procedures.

The first is a proposal in which a temporal 3D model of the surroundings of the robot is created based on information obtained from the robot's sensors during the process of robot operation. This robot system, along with the temporal 3D model, will be discussed in combination with IC.

The second is to utilize a long-term 3D model based on data obtained during the robot operation. Using this long-term 3D model as a simulation environment, we show that it can be applied to learning how to automate robot operations. This proposal describes a method where the robot autonomously navigates using a policy learned

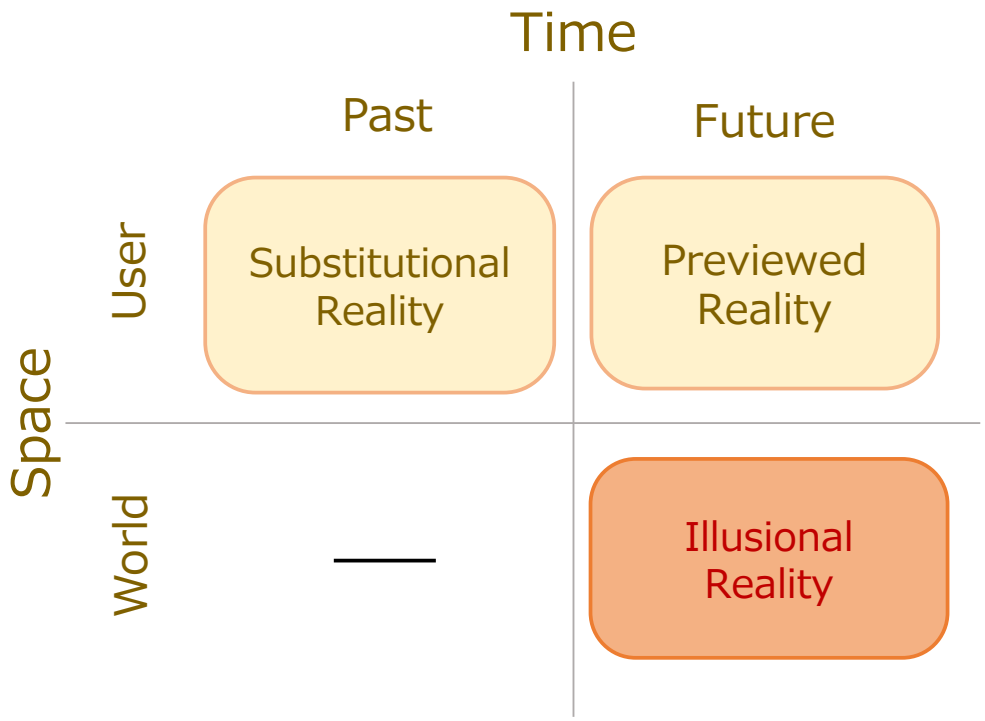


Figure 1.6: The positioning of xR from the view of time and space.

from simulated outcomes within the 3D model. These examples demonstrate how Sim-in-Real can seamlessly integrate model creation into practical operational procedures, enhancing systems' effectiveness and intuitiveness.

The concept of this research is shown in Figure 1.7.

We implemented these approaches in actual mobile robot systems and conducted experiments. The reason for selecting mobile robots as the focus domain is due to the significant contributions that these approaches can offer. Another major domain is arm manipulation, which, although requiring complex operations, has a limited workspace where the amount of environmental change observed by the robot's sensors is relatively minimal. Consequently, the burden of creating virtual spaces for arm manipulation is likely lesser. In contrast, mobile robots experience significant changes in visual information due to their movements, which would require more frequent updates to the virtual spaces. Therefore, the implementation of these approaches in mobile robots contributes more substantially to enhancing user usability.

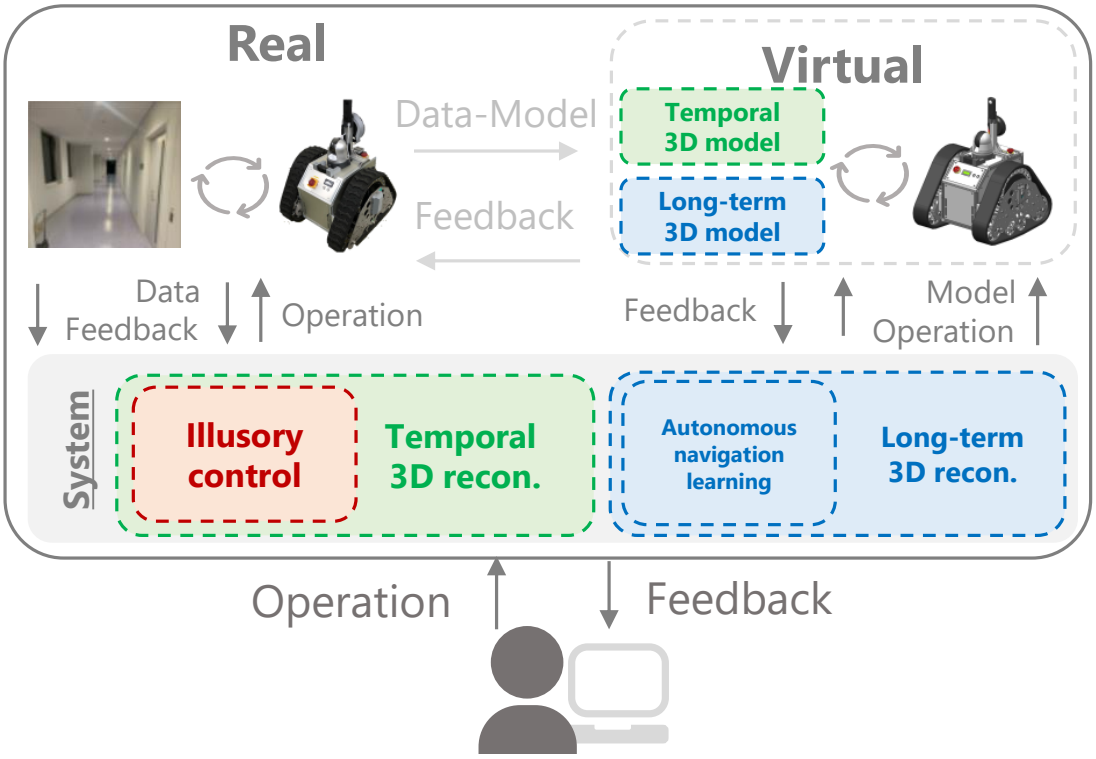


Figure 1.7: The main idea of this research. Illusory Control, shown by a red-colored square, is described in Chapter 2. Temporal 3D reconstruction and model shown by green-colored squares and combination with Illusory Control are described in Chapter 3. Long-term 3D reconstruction and model and autonomous navigation learning shown by blue-colored squares are described in Chapter 4.

1.4 Contributions and Outline

In this research, we address two challenges—User interface and Model construction—to realize AMW. For the user interface, we attempted to solve the issue by implementing IR, where the system seamlessly switches between real and virtual spaces depending on the application. This approach aims to make the interaction between the two spaces intuitive and less cognitively demanding for the user. For model construction, we propose a solution by constructing the virtual space solely from sensor information obtained during the robot operation process. The constructed virtual spaces are then applied to both manual and autonomous robot operations. This method simplifies the creation of virtual spaces without the need for extensive pre-setup or specialized knowledge, enabling more flexible and efficient use of DT in robotics.

In Chapter 2, based on the concept of IR, which aims to switch between real and virtual spaces without making the user aware of the change, we proposed an operation method IC to improve the user interface in teleoperation robots. We conducted a user study using a robotic system that implemented this method. The experimental results indicated that the proposed method could provide a comfortable user experience without compromising efficiency compared to traditional robot operation methods.

In Chapter 3, based on the concept of Sim-in-Real, which involves creating virtual space models without the user's awareness, we proposed a method to construct a temporary virtual space around the robot using only the data observed during robot operations, facilitated by Neural Radiance Fields. We also implemented this method as Instant Illusory Control (Instant IC), thereby realizing a system that requires no prior preparation. Instant IC demonstrated the potential to construct virtual spaces adequate for the uses of IC and suggested further possibilities for enhancing the user experience.

In Chapter 4, continuing with the Sim-in-Real concept, we proposed a learning framework for automating robot operations through a long-term virtual space created solely from data observed during robot operations facilitated by Neural Radiance Fields. Using this framework, we showed that a simulation environment could be built effortlessly from a single instructional image sequence, within which a robot could learn mobility policies. These learned policies showed the potential to be directly transferred to actual robots, highlighting the effectiveness of the approach for practical applications.

Finally, Chapter 5 concludes this dissertation and discusses the future directions.

Notations

The terms used in this dissertation are defined as follows.

- **R-space.** The space in a real-world
- **R-robot.** A robot placed in the r-space
- **R-pos.** The position information of the r-robot
- **R-state.** The state in which the r-robot is the target control of the operator
- **V-space.** A virtual space on the simulator where the 3D model is constructed by scanning the real world
- **V-robot.** The robot placed in the v-space
- **V-pos.** The position information of the v-robot
- **V-state.** The state in which the v-robot is the target control of the operator
- **V-time.** The duration of the v-state
- **IR.** Illusional Relity, as the proposed concept
- **AMW.** Ambient Meta World, as the proposed concept
- **IC.** Illusory Control, as the proposed method
- **EBIAN.** Environmental and Behavioral Imitation for Autonomous Navigation, as the proposed method

2

Robot Teleoperation with Illusional Reality

In this chapter, we discuss the application of IC to robot operations, a concept based on IR, which seamlessly switches between real and virtual spaces without the user's awareness. IC enables users to provide feedback without being conscious of whether they are observing the real or virtual space. This simplifies the complex operations involved in operating robots, offering a more comfortable user experience. Note that the user of the robot system is defined and referred to as the operator or the human operator.

2.1 Introduction

Teleoperated mobile robots have attracted considerable attention with the recent increase in telecommuting and labor shortages. The teleoperated mobile robot is a mo-

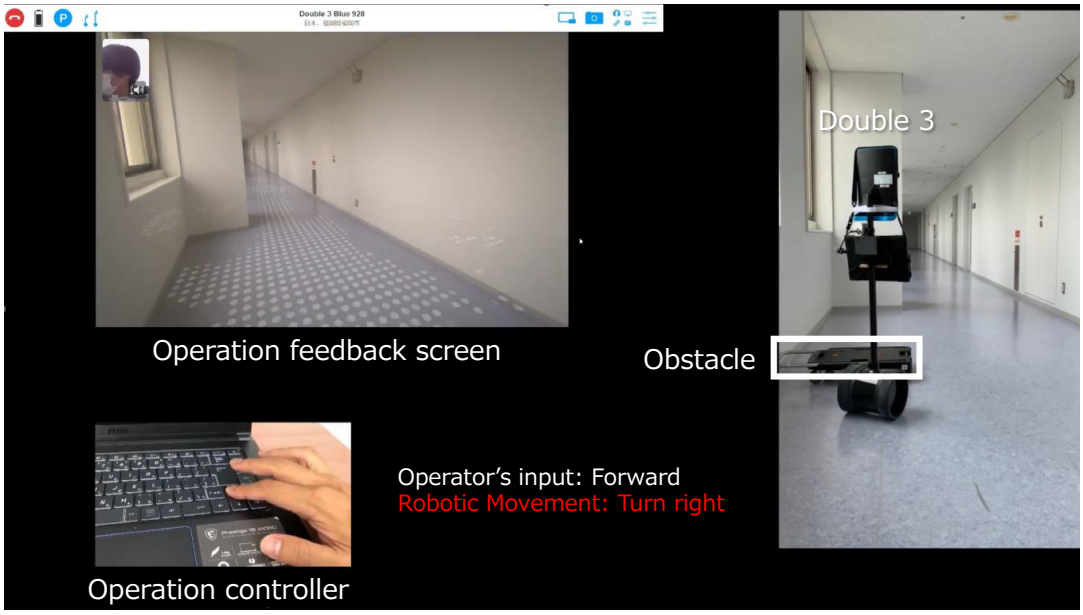


Figure 2.1: Double 3 and its operation feedback screen and operation input controller. The situation in which obstacles are not visible on the feedback screen often leads to discrepancies between operator and robot intentions.

bile communication robot with interfaces such as cameras and microphones. An operator interactively controls the robot through a PC or tablet while receiving images and audio from the camera and microphone of the robot, respectively. Thus, the operators can act through a teleoperated mobile robot as if they were physically present.

However, controlling the remote robot without completely colliding with people or obstacles is challenging. One reason is that the information provided by the remote robot is often insufficient for the operator to understand the robot's surroundings.

Two approaches can be used to address this problem. The first is to increase the information available to the operators and improve their perception of the robot's surroundings. For example, additional information from range sensors might help understand the robot's surroundings better. The second is called shared control (SC), where an operator and an autonomous agent control a single robot while sharing the right of control. SC technology allows robots to detect obstacles using their onboard sensors and autonomously avoid them or stop operation to ensure safe driving. In other words, the robot can autonomously support the lack of the operator's perceptual abilities.

However, while they have the potential to improve teleoperation, the following limitations exist [1]. In the first approach, the difficulty of operation increases as the volume of information increases [32]. Therefore, it is desirable to provide a system

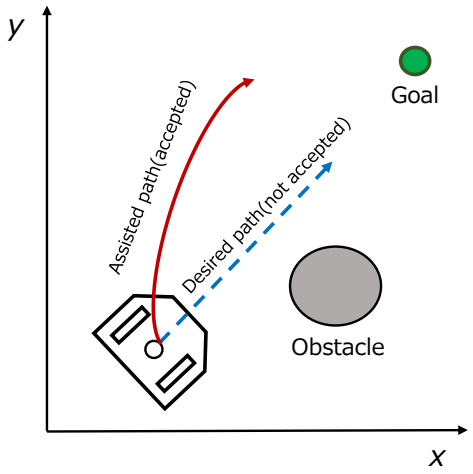


Figure 2.2: In a shared control system, the human operator's intention and trajectory of a robot trying to avoid an obstacle may differ.

that increases the amount of operational feedback information as little as possible. In the second approach, the operator might feel a strong discomfort when there is a disagreement between them and the robot. This is owing to the fact that the motion commands issued by the robot differ from those issued by the human operator, as illustrated in Figures 2.1 and 2.2. In previous work, this has been described as the human operator decisions not matching the robot decisions, which causes the operator to be less accepting of the robot (less trusting in the system) [33]. Additionally, in cooperative behavior between a human operator and robot, discomfort is caused by the disagreement between the robot's and operator's intentions [34]. SC should be utilized because it offers the benefit of accomplishing tasks without requiring complex operations for operators. However, it is desirable to avoid situations where there is a disagreement between the operator and the robot intentions. We compensate for the weakness of existing approaches while adopting their strength. Specifically, we propose a teleoperation method that involves (1) comfortable operation with limited information and (2) no loss of task efficiency compared to SC.

An ideal solution to the problem of diminishing acceptance would be for the robot to avoid obstacles of which the human operator is unaware automatically, while the operator remains unaware of the avoidance behavior itself and feels that the robot is moving as if it were performing the intended actions. However, provided that the robot has autonomy, this is not possible in practice because the feeling that the robot is moving on its own will always arise when the human operator's command and the

robot's action do not match. We propose a method named the *Illusory Control (IC)*, which provides the illusion that the robot is being moved according to the human operator's intention. To create the illusion, we use a digital twin-based cyber-physical system that includes virtual spaces that are copies of the real space, and the robot moves in each space.

The overview and concept of our approach are described in Figures 2.3 and 2.4, respectively. IC involves switching the teleoperated robot world between real and virtual as follows.

- Construct a virtual space, which copies the real space under identical conditions but with no factors inhibiting the operability, such as obstacles.
- Seamlessly switch the operator world from real to virtual while the robot continues to exist at the same position in the real world when issues inhibiting user operability occur in the real world.
- Control the robot from inside the virtual world without operability issues.
- Robot in the real world addresses issues through autonomous movement.
- Seamlessly switch back the operator world from virtual to real when issues are resolved.

Through this system flow, operators can simply convey their intentions to the system, and the robot's autonomy interprets and supports these intentions. Note that IR and IC are not intended for use in operator training; rather, they aim to enable operators to achieve their goals through simple operations.

In addition, we implemented a *virtual feedback rendering* method for the virtual space. The method ensures consistency between real and virtual spaces for comfortable operation.

The proposed method is intended for use when an autonomous agent controls the robot safely. Static obstacles that can be autonomously avoided by the robot are defined as expected obstacles. However, static obstacles that cannot be autonomously avoided by the robot and dynamic obstacles, such as humans, are defined as unexpected obstacles. In addition, if only expected obstacles are in the robot's moving area, the robot is considered to be safe. In the other words, at this stage of this research, IC focuses on only the static environments. However, a situation that has unexpected obstacles

and cannot move safely in the real world is not considered in IC. The current system design for handling unexpected situations is explained in the appendix Chapter B.

The contributions in this chapter are as follows.

- We proposed a teleoperation method *illusory control* that seamlessly switches to operating a robot in virtual space without operability problems only during periods when operability problems may occur with a robot in real space.
- We proposed an image transformation method to switch images between virtual and real spaces without apparent discomfort.
- We proposed a visual effect that the system can control the velocity of a mobile robot with operator discomfort.
- We have verified the practicality of IC system by conducting experiments using actual mobile robots and conducted a user study to evaluate the usability compared to conventional teleoperation methods.

The rest of this chapter is organized as follows: Section 2.2 presents the related work. Sections 2.3 and 2.4 describe the IC and virtual feedback rendering, respectively. A user study illustrating the operation of the proposed method is presented in Section 2.5. Finally, Section 2.6 concludes this chapter.

2.2 Related Work

2.2.1 Teleoperation

Various approaches have been proposed to improve the teleoperation of mobile robots. The two main approaches involve enhancing the operators' ability to perceive remote location information and using the robot's autonomy without relying on the operators' perceptual ability.

Approaches to enhance operators' perceptual ability include telexistence [35], tactile feedback [36], and confirming a situation by expanding the viewpoint [37]. These approaches allow the robot and an operator to share considerable information as if the operator were in a specific remote location, thereby enabling an accurate understanding of the situation and improving teleoperation accuracy. However, there are problems with this approach, such as low operability owing to the increased information volume.

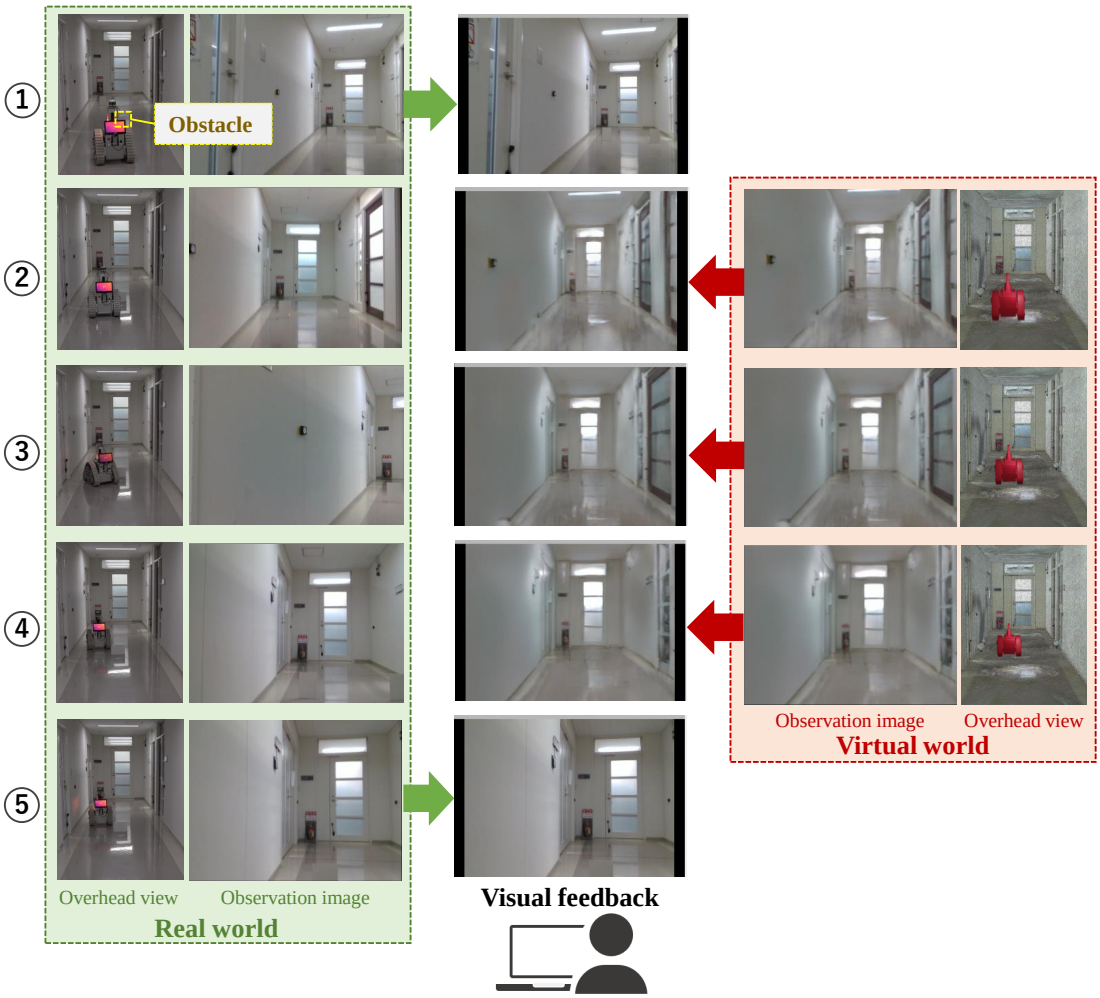


Figure 2.3: IC method. The system is running, and moving tasks are being performed. First, the operator controls the robot in real space (row ①). When the robot in the real space detects that it is about to collide with an obstacle, the robot in the virtual space is placed in the same position as the robot in the real space, and the operator-controlled target is switched from the robot in the real space to the robot in the virtual space (row ②). At the same time, the image feedback to the operator switches seamlessly from the real to the virtual space. In ② - ④, the operator controls the virtual robot in an obstacle-free environment, whereas the real robot moves autonomously in the background to avoid obstacles. In ⑤, the positions of the virtual and real robots are synchronized; thus, the system shows the real space image to the operator, and the operator-controlled target is switched back to the real robot. The supplemental video file shows the system in action.

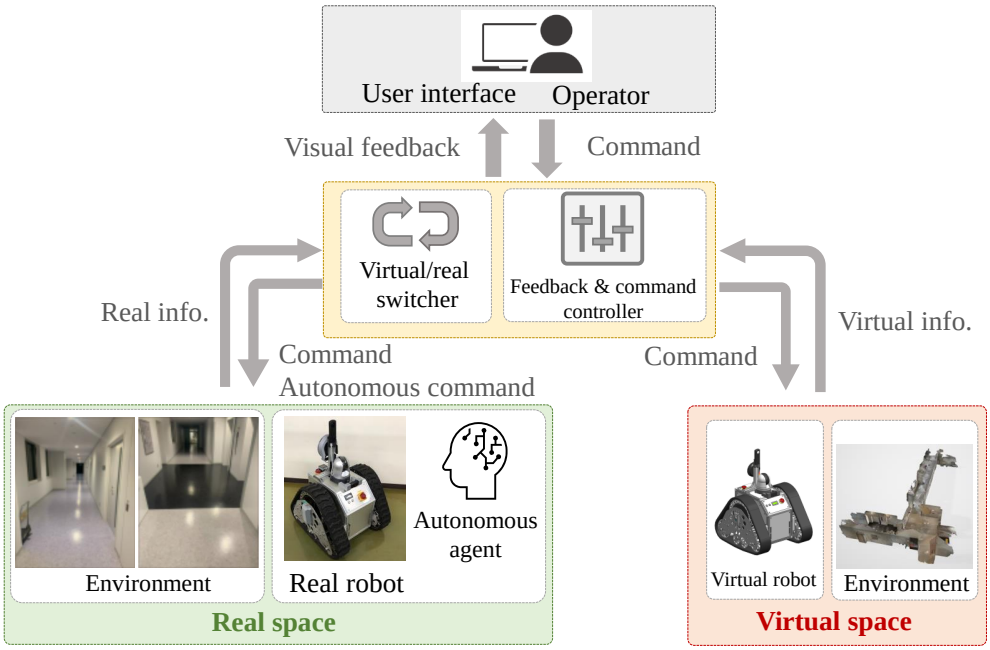


Figure 2.4: Illustration of the IC concept. The concept consists of real and virtual spaces. A 3D model, which is a copy of the real space obtained by point cloud scanning, is placed in a simulator, and the robots in both spaces interact to perform mobile tasks.

In addition, previous research on teleoperation has shown that the wider the viewing angle of the robot's image, the more efficient the work and fewer the collisions, but the operational complexity increases [32].

Several SC approaches are available, such as indicating the navigation path by sketching [38], improving navigation efficiency by probabilistic target estimation [39], and improving operation efficiency by acquiring human-assisted policies through human-in-the-loop reinforcement learning [40]. However, the operator may fail to operate the robot as desired due to the mismatch between the operator and agent intentions because the agent controls the robot preferentially compared to the operator. Previous research has shown that trust influences operator decision time in operator-agent interactions [33]. In other words, a disagreement between the operator and agent's intentions yields less trust in the system and increases the operator's decision-making time. Some approaches have been proposed to enhance mutual understanding between the operator and agent, such as visualizing the agent's intentions up front [41]. However, according to a study evaluating operators' impressions of robot operation assistance functions, operators tend to prefer high controllability over

robots to task automation [42]. Therefore, we focused on directions that provide operators with a high degree of controllability. Previous research has been dominated by approaches [41] [31] that visualize the possibility of robots behaving contrary to human expectations. However, the IC approach, which ensures high controllability by hiding robot movements that are contrary to human expectations, has not been studied extensively.

Predictive displays have been studied primarily to improve operability during communication delays [43, 44, 45, 15, 31]. There are methods that display future poses on a local map, using image transformation to apply position and scale [44], and AR/VR [31] [45] [15]. Predictive displays allow operators to change their behavior based on future system predictions. In contrast, our method changes the operation target space so that the operator does not change their behavior while not being aware of the change. Moreover, the main challenge of predictive displays is interpolating the difference between human expectations and the robot's current state under delays of several seconds. Our method takes care of the delay by introducing the virtual feedback rendering method, as described in Section 2.4.

In industrial robotics, AR/VR-based control methods have been studied in the domain of arm manipulation [46] [47]. VR-based arm manipulation works well for control in specific structured environments. However, unexpected obstacles can appear in mobile robots, especially in unstructured environments. In the domain of mobile robots, which involves dealing with such unexpected situations, it is difficult to complete the feedback method to the operator only with VR. Recently, technologies that can be applied to unstructured environments have been studied, and their applicability to mobile robots is expected to increase [48]. However, this study focused on a desktop application-based teleoperation method that switches between real and virtual worlds, which is closer to practicality.

2.2.2 Virtual Feedback Rendering

Determining the v-space is an important factor influencing operability. Several methods, such as texture mapping, are used to seamlessly transit the appearance between the r-space and v-space in VR literature [49, 50]. However, we focus on image transformation using neural networks to dynamically absorb environmental changes such as lighting conditions and weather in the r-space in v-space rendering. Neural network-based methods include style transfer [51] and GAN-based methods to learn correspon-

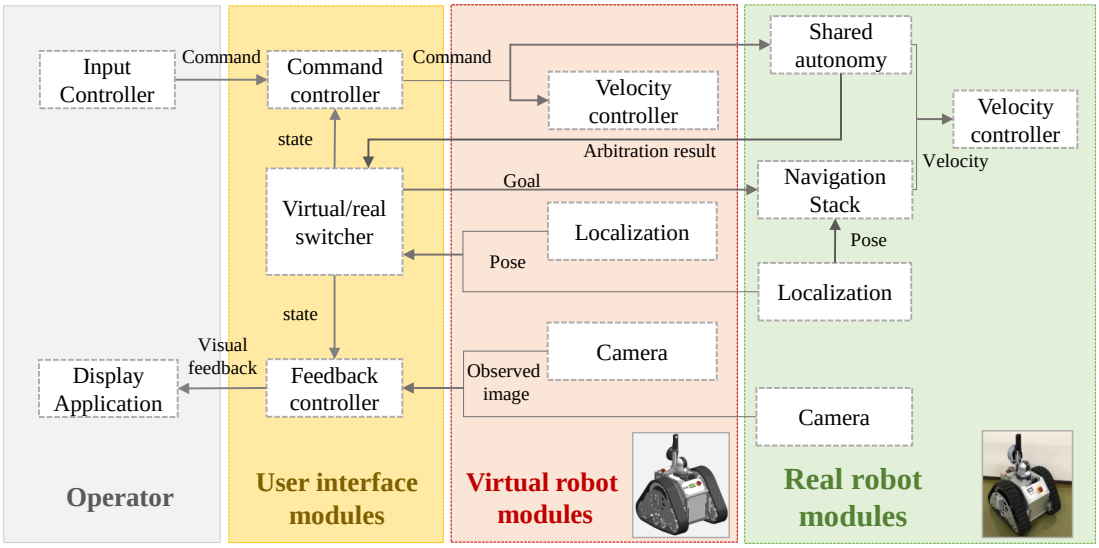


Figure 2.5: System architecture of IC.

dences between images [52, 19]. In addition, the simulator environment uses point cloud information as the input and outputs the corresponding real image [53]. In the proposed system, we implemented a virtual-to-real image transformation function using CycleGAN [19]. CycleGAN can transform the images in real-time. It is used as the base of a sim-to-real transfer method for autonomous movement [6, 20], and requires a small amount of data to learn the transformations between image domains. It is possible to use methods other than CycleGAN, such as [54], which corrects geometric inconsistencies and generates smooth surfaces in the proposed system.

Previous studies have shown that visual effects can induce behavioral changes in the operator [55, 56]. In addition, the optical flow in peripheral vision affects the sense of speed among self-motion senses [57]. The proposed system introduces a visual effect method as part of the virtual feedback rendering to affect the operator's sense of speed, as detailed in Section 2.4.2.

2.3 Method

2.3.1 System Overview

The concept of the IC is shown in Figure 2.4. The IC system consists of the r-space, v-space, and the robots in each space. The v-space is a simulator with a 3D model of the r-space constructed using a 3D reconstruction method with a 3D scanner. The robots

interact with the environment in each space to perform mobile tasks. In the v-space, no objects interfere with the robot's movement. The r-robot is remotely controlled by the operator, similar to teleoperation methods in IC. When the r-robot is about to collide with an obstacle, the operator-controlled target is switched from the r-robot to the v-robot, and the operator controls the v-robot in the v-space. Meanwhile, the r-robot performs autonomous avoidance of the obstacle outside the operator's line of sight. After completing the autonomous obstacle avoidance, the control target is switched back to the r-robot.

The overall module structure of the proposed IC framework is illustrated in Figure 2.5. User interface modules regulate the operator commands and provide feedback images to the operator. The r-robot and v-robot modules control the r-robot and v-robot, respectively.

2.3.2 Module Responsibilities

2.3.2.1 User Interface Modules

The user interface modules consist of a command controller, virtual/real switcher, and feedback controller. The command controller receives operator input and issues movement commands to either the virtual or real robot modules, depending on the state. The state is determined by the virtual/real switcher. The virtual/real switcher switches the state from r-state to v-state based on the arbitration result of the real robot's behavior by shared autonomy. The virtual/real switcher periodically monitors the posture information of both the virtual and real robots and switches from v-state to r-state. The details of state switching are described in Section 2.3.3. The feedback controller receives observation images from both cameras of virtual and real robots and switches feedback images according to the state of the virtual/real switcher. The r-space and v-space observation images are switched by 0.3 seconds cross-fade. Because there is a slight error between v-pos and r-pos, switching images without the cross-fade can cause the operator to feel discomfort, as if they suddenly teleported. Cross-fade was adopted to reduce this discomfort.

2.3.2.2 Real and Virtual Robot Modules

Firstly, in Chapter 2, it is assumed that a map is prepared where the scales of the real and virtual spaces are matched. The localization results can be shared between the real

and virtual spaces without any transformations.

The virtual robot modules consist of a velocity controller, localization, and camera. The velocity controller receives operation commands consisting of translational velocity υ [m/s] and rotational velocity ω [rad/s] and sends its operation commands to the v-robot model. The camera sends the images obtained from the v-robot's camera model, and the localization performs odometry estimation. In the system, odometry estimation is performed by synthesizing the wheel odometry of a two-wheeled mobile robot and LiDAR odometry using an extended Kalman filter. Odometry estimation methods can be replaced by other methods depending on the environment and domain.

The real robot modules consist of a velocity controller, localization, camera, shared autonomy, and navigation stack. The velocity controller, localization, and camera are the same as those of the virtual robot modules. The shared autonomy receives operation commands and calculates the future trajectory using the base local planner of ROS; it determines whether the calculated future trajectory will reach the obstacle; thereafter, it is sent as an arbitration result. The navigation stack is provided as a ROS package. Considering a goal pose, it performs autonomous movement toward this goal.

2.3.3 Switch State between V-state and R-state

2.3.3.1 Real-to-virtual

This section describes the flow of switching from the r-state to the v-state.

The operator starts the operation based on the camera image of the r-robot, which is the control target. The operator controls the robot using the input controller. Shared autonomy notifies arbitration results based on operation command to the virtual/real switcher. When notified of the arbitration result that the future trajectory is to reach an obstacle, the virtual/real switcher switches from the r-state to the v-state. Then, the v-robot is moved to the same position as the r-pos in the v-space. Simultaneously, images shown to the operator are switched to the virtual images while in the v-state.

2.3.3.2 Autonomous Navigation

As shown in Figure 2.6, the r-robot moves autonomously with the v-pos as a sub-goal during the v-state. The v-robot sends v-pos to the r-robot, and the r-robot starts the autonomous movement with the same position as v-pos in r-space as its sub-goal ((A)①). While the r-robot is moving autonomously, the v-pos changes as the v-robot

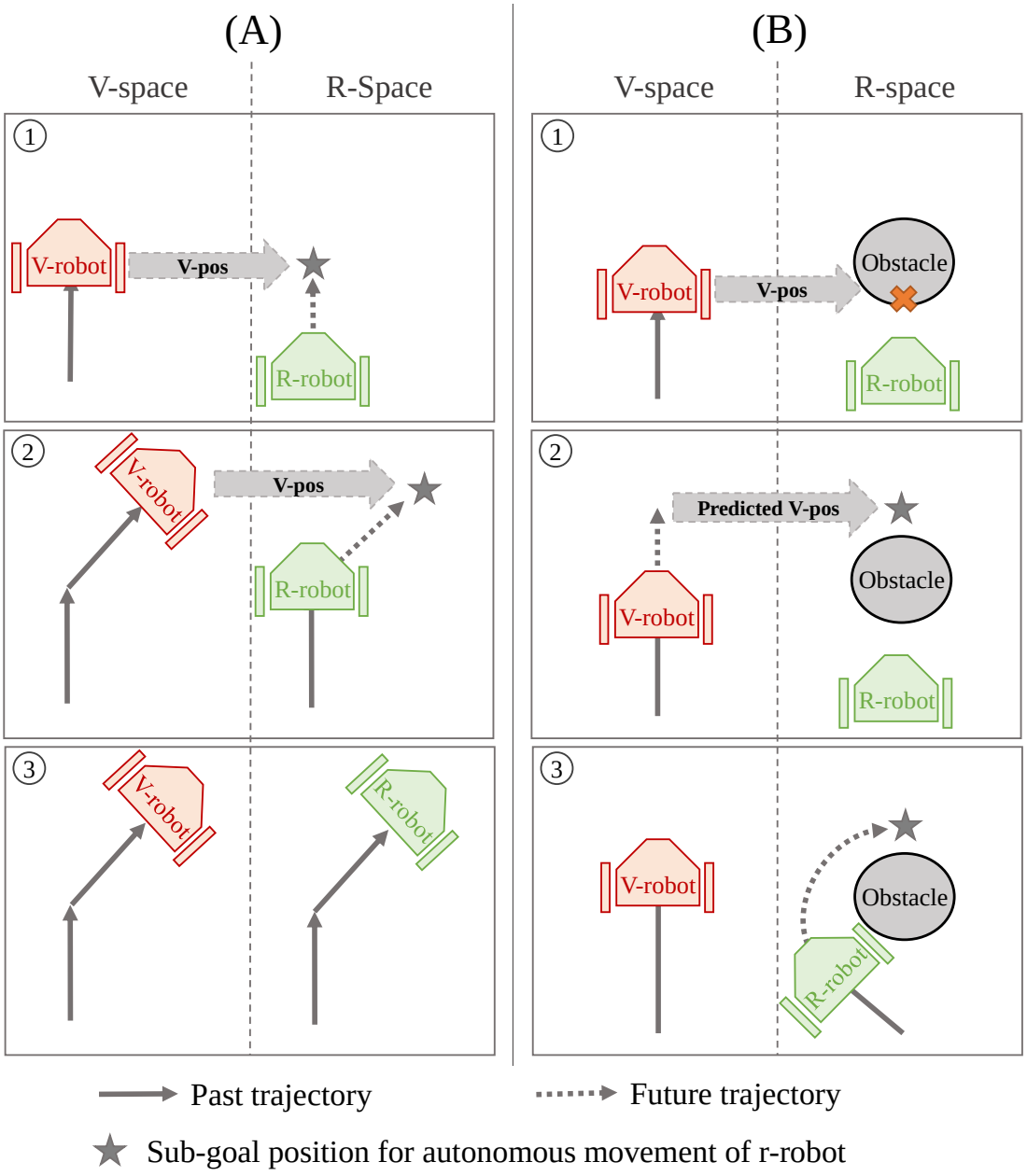


Figure 2.6: Robot navigation. In column (A), the r-robot moves autonomously to follow the v-pos that the operator controls. In column (B), a valid point is set in the future trajectory of the v-robot as a sub-goal if there is an obstacle at the v-pos.

is moved by the operator. If the distance between the previously set sub-goal of the r-robot and current v-pos exceeds the threshold value, a new sub-goal for the r-robot is set ((A)②). When the distance between v-pos and r-pos becomes less than the threshold value, the system determines that the positions of v-robot and r-robot are synchronized and then terminates the autonomous movement of r-robot((A)③).

As shown in Figure 2.6 (B), if there is an obstacle in the same position as v-pos in r-space, the future trajectory of the v-robot is calculated, the corresponding cost is obtained, and the sub-goal is set to a point where the cost is less than a certain level. If there is an obstacle in the same position as the v-pos in the r-space, the sub-goal for the autonomous movement of the r-robot is not set ((B)①). Based on the operator's input command [υ, ω] to the v-robot, future trajectories are calculated for a pre-determined number of seconds. The positions on that trajectory are sampled in sequence at equal intervals, beginning with the position closest to the robot. The cost of the sampled position is obtained from the local map; if the cost is less than the threshold, the position is set as the sub-goal of the r-robot ((B)②). The r-robot begins autonomous movement toward the set sub-goal ((B)③).

2.3.3.3 Virtual-to-real

In the following, we describe the flow of control return from the v-state to the r-state.

The virtual/real switcher can acquire v-pos and r-pos. The feedback image to the operator is switched from v-space to r-space when the difference between the respective positions becomes less than a threshold value. Simultaneously, the state is switched from the v-state to the r-state. V-pos and r-pos are assumed to represent the same world coordinate system, using the position [x, y] and angle θ between the x-coordinate and robot, represented by $[x_v, y_v, \theta_v]$ and $[x_r, y_r, \theta_r]$, respectively. The state s is determined using the distance $d_{vr} = \sqrt{(x_v - x_r)^2 + (y_v - y_r)^2}$ and orientation difference $\theta_{vr} = |\theta_v - \theta_r|$ between the v-robot and r-robot. In summary, the v-state and r-state follow the following equation:

$$s(d_{vr}, \theta_{vr}) = \begin{cases} r - state & (d_{vr} < d_{th} \; and \; \theta_{vr} < \theta_{th} \;) \\ v - state & (otherwise), \end{cases} \tag{2.1}$$

where d_{th} and θ_{th} are the threshold parameters.

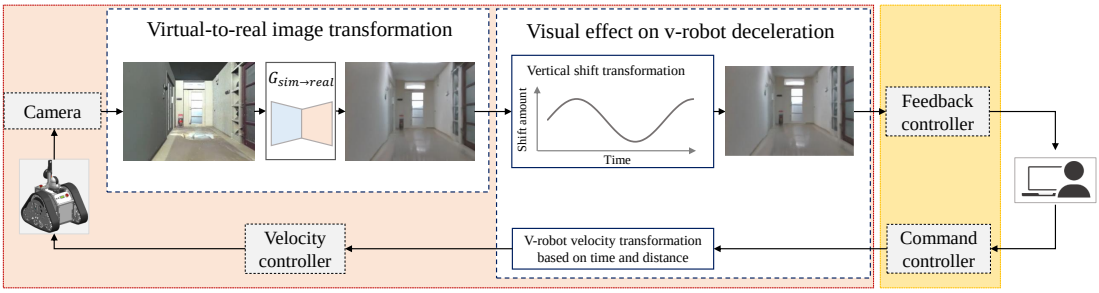


Figure 2.7: The architecture of virtual feedback rendering.

2.4 Virtual Feedback Rendering

In the proposed system, the operator controls the v-robot or r-robot, depending on the situation. It is desirable to ensure the consistency of v-space and r-space for comfortable operation. Two main issues need to be addressed to ensure consistency. We attempt to address these two issues using a virtual feedback rendering method. The architecture is shown in Figure 2.7. Processes of virtual-to-real image transformation and visual effect on v-robot deceleration are added to the virtual robot modules shown in Figure 2.5. The details of this process are described below.

2.4.1 Virtual-to-real Image Transformation

The first issue is that the appearances of the v-space and r-space considerably vary. When meshing is performed from the scanned point cloud information, some parts of the model and color tone may differ between the v-space and r-space (Figure 2.8 (A)(B)). Therefore, the operator may feel uncomfortable because the texture of the environmental information feedback is significantly different when the state is switched from r-state to v-state. We hypothesized that the ability to transition from r-space to v-space without discomfort would reduce the operator's workload in the teleoperation of robots. The texture of the visual information in the v-space was made to resemble that of the r-space using CycleGAN [19] to bring the images on the simulator closer to the r-space images. The results of the image transformation are shown in Figure 2.8 (C). The computation time for image conversion using Nvidia Geforce RTX 2080 Ti is 60-80 ms.

2.4.2 Visual Effect on V-robot Deceleration

The second issue is that v-pos and r-pos gradually become farther apart. A possible solution to this problem is to force the robot to stop regardless of the operator's control commands. However, this may make the operator feel as if they cannot control the robot as desired because of deceleration, which may result in strong discomfort. In our previous study, we addressed this issue [1] by applying visual effects that the operator would not notice, shown in Chapter A in appendices. We guided the operator unconsciously so that the difference between the v-pos and r-pos would decrease over time. Specifically, the v-robot gradually decelerates in the v-space while the visual effect prevents the operator from experiencing this deceleration.

First, as the difference between the current v-pos and r-pos d_{vr} increases, with increasing v-time t, the v-robot velocity gradually decelerates compared to the operator input velocity v_{org} as follows:

$$v(t) = \begin{cases} \frac{v_{org}}{\alpha d_{vr} + \beta t + \gamma} & (t \geq t_{th}) \\ v_{org} & (t < t_{th}), \end{cases} \tag{2.2}$$

where α, β, and γ are the coefficient parameters, and t_{th} is the threshold parameter.

However, decelerating the robot may also cause discomfort to the operator. Therefore, we implemented a method to reduce the deceleration feeling of the robot based on the visual effect. It creates visual motion by gently oscillating the camera image of the robot (Figure 2.9). The visual effect moves in a constant cycle, regardless of the v-robot velocity, to avoid distracting the operator by dynamically changing the amount of vertical vibration.

2.5 User Study

The usefulness of the IC was verified through quantitative and qualitative user studies ($N = 19$, 18 male and one female) with ages ranging from 21 to 55. Three participants were university researchers specializing in robotics, 15 participants were students specializing in robotics, and one participant was an administrative staff member who did not specialize in robotics. These participants were randomly gathered, regardless of their remote control skills, and were not compensated. The experiments were approved by the Ethics Review Committee of the Graduate School of Information Science and Electrical Engineering at Kyushu University.

(A) V-space image (B) R-space image (C) Virtual-to-real transformed image

Figure 2.8: (A) Image obtained from the camera mounted on the v-robot. (B) Image of the actual r-space. (C) Result of the virtual-to-real image transformation using the (A) as input.

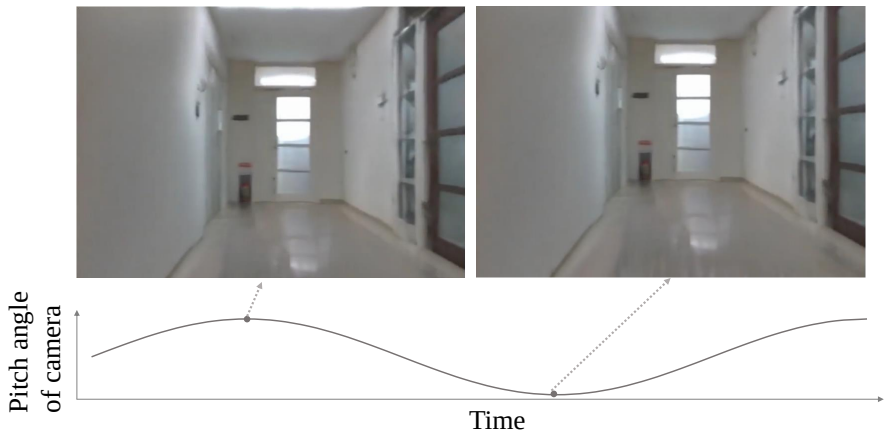


Figure 2.9: View of the visual effect in action. The feedback image oscillates over time.

2.5.1 Methods

We conducted 4×1 user studies that compared the proposed method with conventional methods. A summary of the methods is given below:

- **Direct teleoperation (DOP):** The robot moves according to the directional input from the operator. There is no capacity to detect obstacles and autonomously stop or avoid them.
- **SC:** The directional keys of the operator and local cost map are used as the input. The local planner of the ROS is used to calculate a path that avoids obstacles and autonomously turns the robot in a direction free from obstacles.
- **SC with alert (SCA):** In addition to the SC function, the system notifies the participant by displaying an alert when the input command is changed in a direction without obstacles. The alert only notifies the operator that the robot cannot proceed in the direction indicated.
- **Proposed IC method:** This is the proposed method described in Section. 2.3 and Section. 2.4.

2.5.2 Task

The participants sat in front of a PC display, watched visual feedback, and operated the mobile robot using a PlayStation controller in Figure 2.10 (A). Each teleoperation method was implemented as a desktop application. The method can be implemented as a VR application using a VR headset. However, the desktop application was used because the operating environment was still based on a display and controller. Additionally, there were concerns that participants would feel uncomfortable about the slight shift in the robot's position when the participant's view changed between v-space and r-space when using a VR headset, as well as concerns about motion sickness caused by the visual effects of vertical vibration. The robot started from the starting point, as shown in Figure 2.10 (B), and aimed for the goal point while avoiding obstacles. The task was performed once per method. To avoid teleoperation skill improvement from affecting the experimental results, the order of the four teleoperation methods was randomized for each participant. After each task, the participants answered a questionnaire.

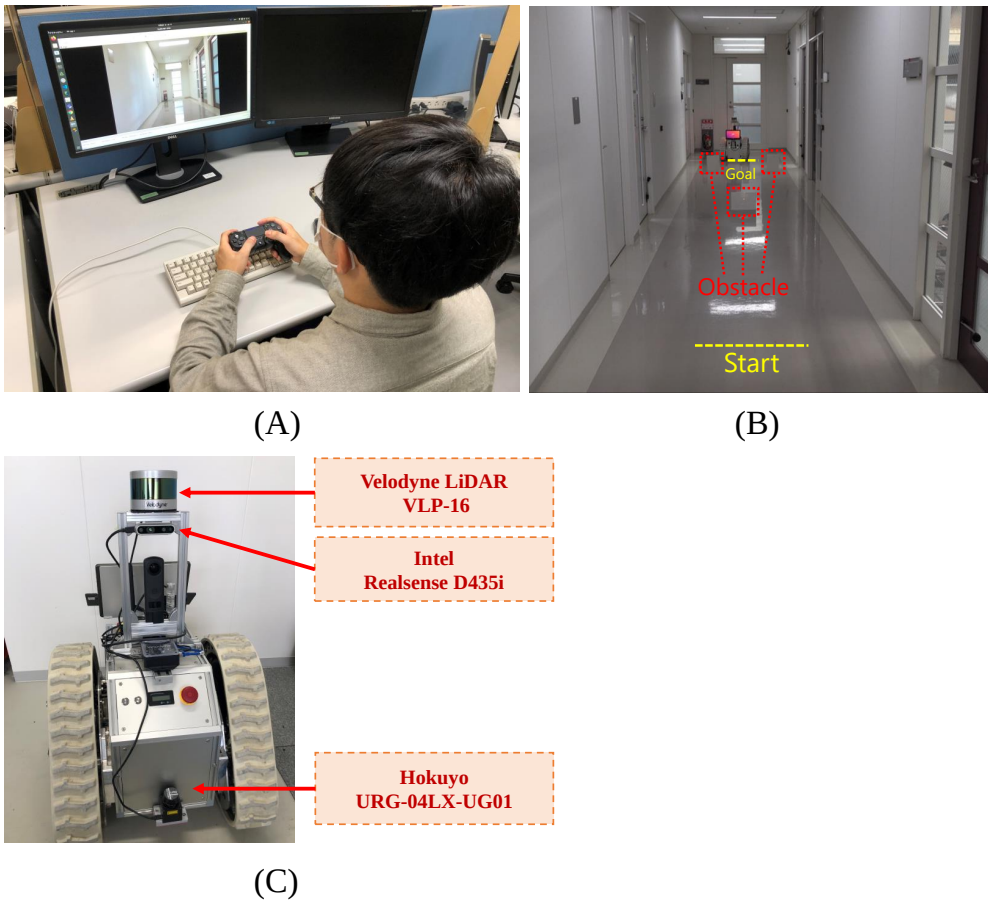


Figure 2.10: (A) Participant performing teleoperation. (B) Experimental course. (C) The robot has a camera (Realsense D435i) for image feedback to the operator, LiDAR (Velodyne VLP-16) for localization on a global map, and LiDAR (Hokuyo URG-04LX-UG01) for obstacle recognition around the robot.

2.5.3 Hardware and Environment

The mobile robot used in the experiment was a crawler-type non-holonomic robot, as shown in Figure 2.10 (C). The translation and rotational velocities were 0.3 m/s and 0.5 rad/s, respectively. The velocity should be constant. A RealSense D435i camera was mounted on the robot at approximately 0.7 m from the ground. The camera had a horizontal and vertical field of views of 69.4 and 42.5°, respectively. As shown in Figure 2.10 (B), three obstacles were installed in the experimental course, two of which were 0.3 m high and one 0.2 m high. Therefore, when the distance between the robot and the obstacle approaches approximately 1.0 m, the obstacle cannot be perceived. Thus, the operator was asked to operate in a situation where it became difficult to perceive the obstacle at a certain point. The robot body had a radius of 0.4 m. The aisle width was 2.0 m in wide areas and 0.9 m in narrow areas between obstacles. The only feedback to the participants is the camera image, and there is no non-visual feedback such as audio or vibration.

The Wi-Fi router used in the experiment was an ASUS AX5700, which connected the client PC to the mobile robot's PC in the 5 GHz bandwidth. Communication was conducted within the intranet, not via the Internet. Note that the mobile robot and teleoperation environment are located separately outside and inside the room, respectively. In addition, participants cannot directly observe the mobile robot moving or directly hear the sounds it makes as it moves. Latency was 0.45 seconds on average. Thus, the impact on the task performance is small in a 2-DOF system [58]. Latency measurements were based on video taken at 30 fps to ensure that the participant, controller, and display could be observed. The measurement started at the frame when the participant entered a command with the controller and ended at the frame when the robot started moving on the feedback video on the display. The result is the mean of the above measurements taken 20 times (5 each of up, down, left, and right commands), separately from the experiment with the teleoperation task.

2.5.4 Measurements

In this experiment, we formulate the following hypotheses: **"H1. The IC method offers comfortable operation with limited information"**, **"H2. The IC method has no loss of task efficiency"**. For **H1**, comfort was measured by comparing the results of responses to a questionnaire. Note that, in this user study, we focused on subjective

evaluations, such as task load and comfort, but not on the user interface evaluations by metrics, such as system usability scale [59]. Because we want to evaluate the differences in impressions based solely on the control methods, without being influenced by the design of the user interface and user experience.

1. **Workload (NASA-TLX [60]):** This is a measure of the participant's workload. There are six questionnaire items: mental demand, physical demand, temporal demand, performance, effort, and frustration level. Each item was rated on a 10-point Likert scale from 0 to 10. A Japanese translation of this scale was used for the evaluation [61]. We hypothesized that IC would score lower than SC and SCA. In scenarios with limited visual information, the robot is more likely to perform actions that are contrary to the participant's intentions, increasing the workload.

2. **Controllability:** This measure determined whether the participants were comfortable operating the robot. A questionnaire item checked whether the participant could operate the robot to their expectations. The questionnaire items were evaluated using a seven-point Likert scale. We hypothesized that IC would score higher than SC and SCA because the IC method switches from r-state to v-state in situations where the robot does not move as desired.

3. **Relief:** This measure determined the relaxation level of participants while operating the robot. A questionnaire item asked whether the participant could operate the robot without paying attention to obstacles. The questionnaire items were evaluated using a seven-point Likert scale. We hypothesized that IC would score higher than DOP, SC, and SCA because, in IC, the system switches the target of operation to an ideal v-space with no obstacles when an obstacle is detected.

For **H2**, task efficiency was measured by comparing the following items:

- **Task execution time:** This is the time taken for the mobile robot to reach the goal point from the time it left the start point.

- **Obstacle collision:** The number of times the robot collided with obstacles

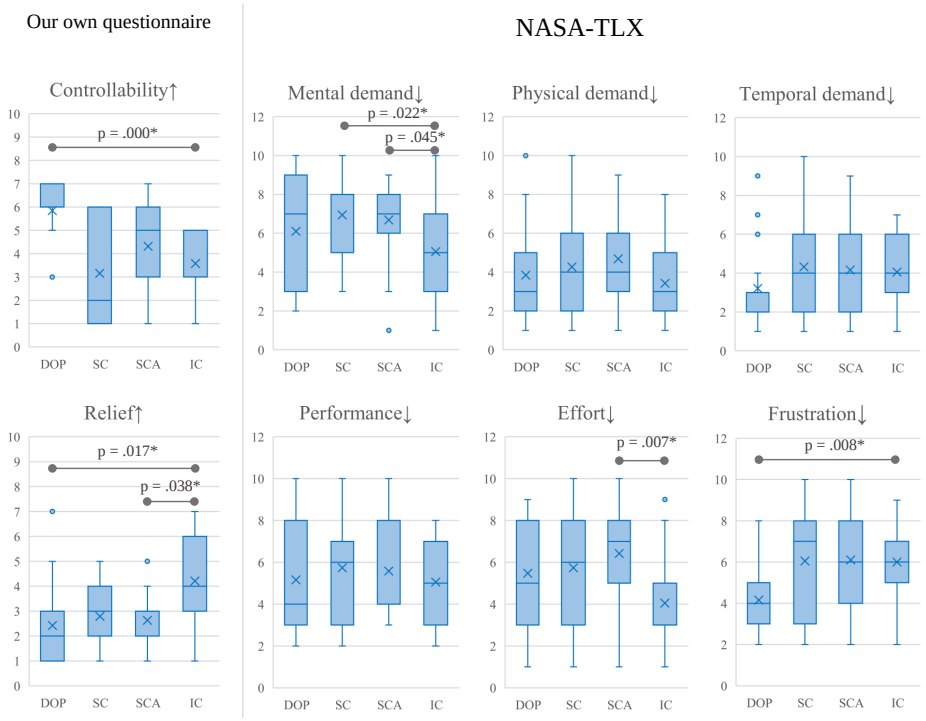


Figure 2.11: Results for comfort by subjective measures. The post hoc Steel test was used to compare IC with the conventional method; (*) denotes $p < 0.05$.

2.5.5 Results and Discussion

2.5.5.1 Comfortable Operating under Limited Information

The results of the comfort evaluations are presented in Figure 2.11 and Table 2.1. First, the normality of each questionnaire result shown in Figure 2.11 was tested using the Kolmogorov-Smirnov test. In this case, since the results included results with non-normality, nonparametric Kruskal-Wallis and post hoc Steel tests were performed. In the steel test, IC is the control group. The significance level for each test was 0.05.

1. **Workload:** **H1** (1) stated that IC scores would be lower than SC and SCA. Kruskal-Wallis test indicated significant differences across the conditions in mental demand ($\chi^2 = 7.82, p = 0.049$), effort ($\chi^2 = 8.79, p = 0.032$), and frustration ($\chi^2 = 10.08, p = 0.018$) included in the NASA-TLX. The post hoc Steel test also showed significant differences in mental demands between SC and IC ($p = 0.022$) and SCA and IC ($p = 0.045$), effort between SCA and IC ($p = 0.007$), and frustration between DOP and IC ($p = 0.008$). Therefore, **H1** (1) was only partially supported.

Table 2.1: Result for comfort by unaccepted command rate

	Unaccepted command rate [%]
DOP	N/A
SC	29.94±15.64
SCA	13.44±9.03
IC	N/A

Table 2.2: Results of task efficiency

	Task execution time [s]	Obstacle collision [Times]
DOP	49.58 ± 9.43	1.05 ± 0.60
SC	103.37 ± 102.68	0.37 ± 0.48
SCA	76.21 ± 41.49	0.37 ± 0.58
IC	84.58 ± 29.30	0.16 ± 0.36

IC was superior to SC and SCA in meeting mental demands. Moreover, IC was superior to SCA in terms of effort. The improved performance could be due to the difference in the amount of judgment in obstacle avoidance between IC and SC/SCA. IC switches from r-state to v-state in situations where the participant must watch out for obstacles. The participant's input is carried over to v-space, where they need to make no decisions on obstacles. In contrast, SC and SCA directly feedback that the participant's input was not always ideal in situations requiring obstacle avoidance. Therefore, the participant must repeatedly rethink and reenter the input to find a path that successfully avoids the obstacle. As shown by "Unaccepted command rate" in Table 2.1, between the start and end of the task, there was a disagreement between the participant and robot $29.94 \pm 15.64\%$ and $13.44 \pm 9.03\%$ of the total command input time for SC and SCA, respectively.

However, DOP has a better score than IC in terms of frustration. This is because, in DOP, participants could operate the robot without assistance in autonomously avoiding obstacles. However, it is difficult to avoid collisions with obstacles when only limited visual information is available, and most participants fail to notice the collision with the obstacle. Therefore, as shown in Table 2.2, DOP has the highest number of collisions.

2. **Controllability:**

   H1 (2) stated that IC would score higher than SC and SCA. Kruskal-Wallis test indicated significant differences across the conditions in the controllability ques-

tionnaire item ($\chi^2 = 23.45, p = 0.000$). The post hoc Steel test also showed significant differences between DOP and IC ($p = 0.000$). Therefore, **H1** (2) was not supported.

DOP scores better than IC on the question "whether the participants feel that they could operate the robot to their own expectations," for similar reasons as in H1 (1).

3. **Relief:**

   H1 (3) stated that IC would score higher than DOP, SC, and SCA. Kruskal-Wallis test indicated significant differences across the conditions in relief questionnaire item ($\chi^2 = 10.36, p = 0.016$). The post hoc Steel test also showed significant differences between DOP and IC ($p = 0.017$) and SCA and IC ($p = 0.038$). Therefore, **H1** (3) was partially supported.

   The results support that IC was superior to DOP and SCA. Thus, switching the target of operation to an ideal v-space without obstacles helps the operator relax during the task.

2.5.5.2 No Loss of Task Efficiency

The results of the task efficiency evaluation are shown in Table 2.2. IC approach was found to be as task-efficient as conventional methods.

In this scenario, DOP had the shortest task execution time, followed by SCA, IC, and SC. In Figure 2.12, IC showcases with and without a smooth position synchronization. The task execution time is 49 seconds with smooth position synchronization. Therefore, it is expected that the IC task execution time can be improved to the equivalent of DOP by improving the parameters for autonomous movement and future advances in autonomous movement technology.

Additionally, IC had the fewest number of collisions in the r-space, followed by SC/SCA and DOP. While DOP had the shortest task execution time, it also had the most collisions with obstacles. This is because many participants failed to realize the collision with an obstacle and continued with the advancing operation. Although SC, SCA, and IC are designed to avoid obstacles autonomously, there were times when the obstacle entered the blind spot of the LiDAR sensor on the mobile robot. The blind spot can be avoided by adjusting the parameters of autonomous moving and adding more sensors to the mobile robot.

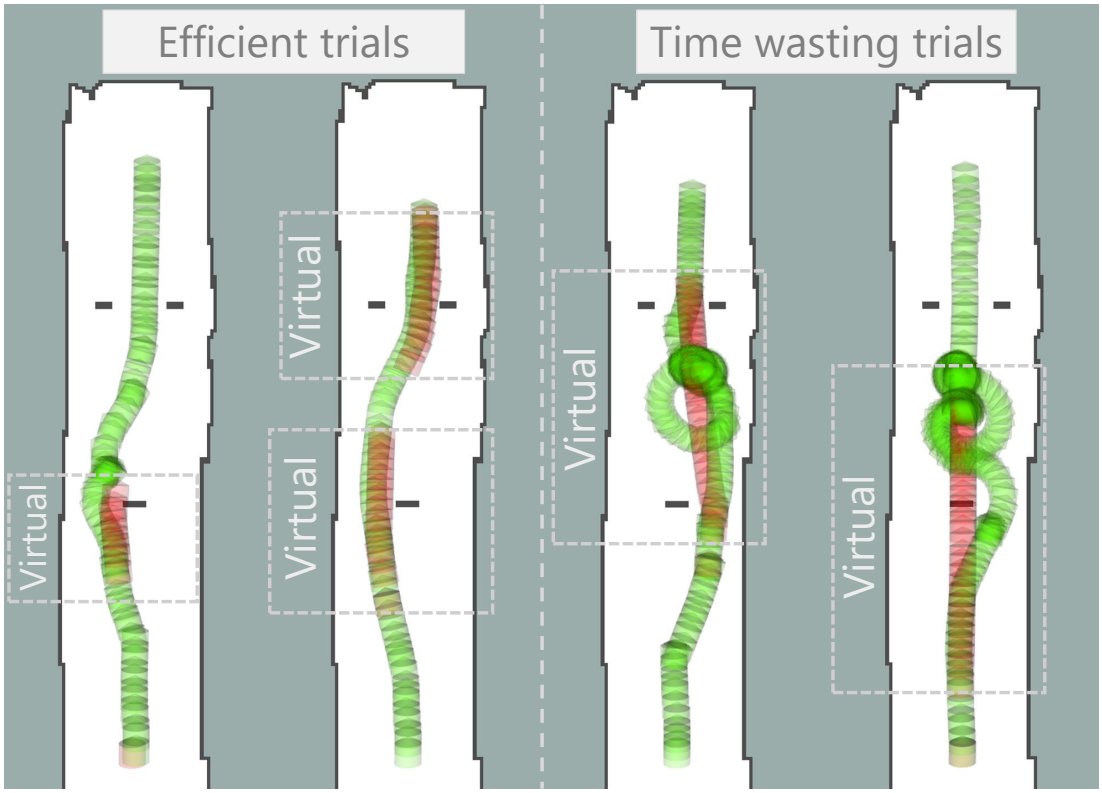


Figure 2.12: Examples of cases where the navigation of the r-robot was completed efficiently and took a long time in IC. The green and arrows show the trajectories of the r-robot and v-robot, respectively. In the region of the v-robot trajectory, the operator operates the v-robot and watches the images in the v-space.

2.5.5.3 Ablation Study

An ablation study was conducted to evaluate the effectiveness of the function that reduces the differences in appearance and positions during the switch between v- and r-spaces.

Section 2.4 described virtual feedback rendering, and two issues are outlined. First, the appearances of the v-space and r-space vary, and second, v-pos and r-pos gradually become farther apart. This subsection describes the effects of virtual feedback rendering on these issues. We verified the improvement effect by comparing IC without image transformation using CycleGAN (IC (-GAN)), without visual effect (IC (-VFX)), and IC with both. IC (-GAN) and IC (-VFX) were added to the methods of the teleoperation experiments described in Section 2.5 and evaluated in the same way.

Figure 2.13 showed the result of comfort by subjective measures. First, the normality of each questionnaire result shown in Figure 2.13 was tested using the Kolmogorov-Smirnov test. Because the results included results with non-normality, nonparametric Kruskal-Wallis and post hoc Steel-Dwass tests were performed. Kruskal-Wallis test indicated no significant differences across any conditions. Table 2.3 showing the result of task efficiency reveals no significant differences in task time and the number of obstacle collisions.

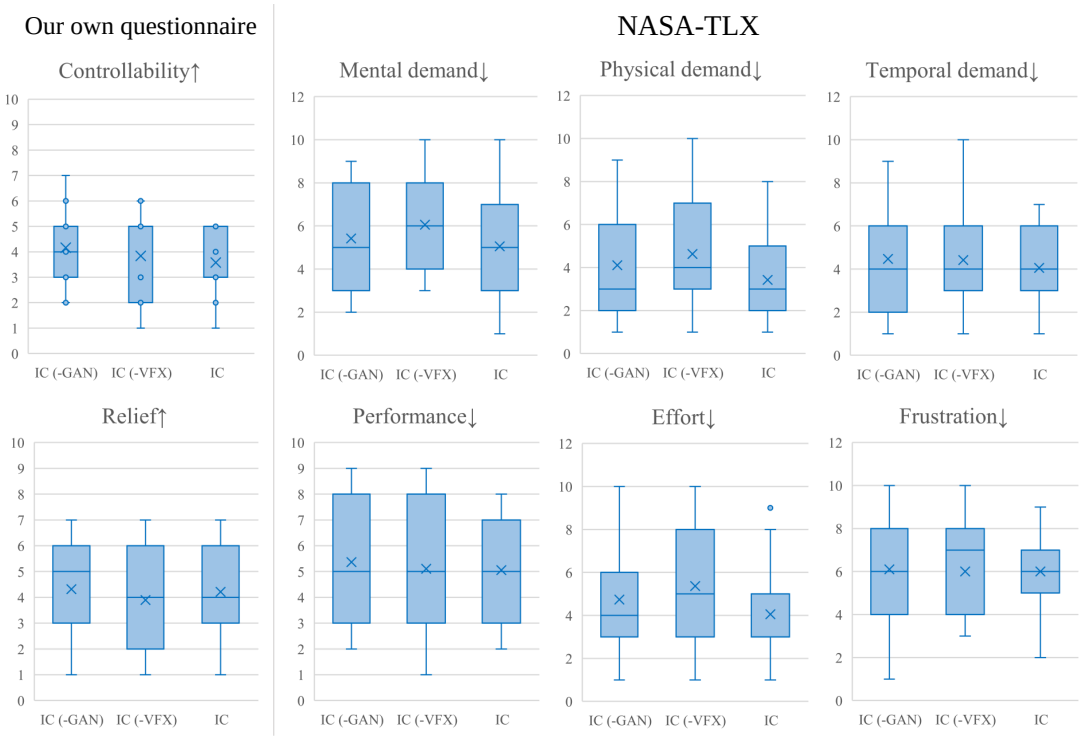


Figure 2.13: Results for comfort in the ablation study

Table 2.3: Results for task efficiency in the ablation study

	Task execution time [s]	Obstacle collision [Times]
IC (-GAN)	84.63 ± 34.74	0.21 ± 0.41
IC (-VFX)	89.84 ± 37.46	0.21 ± 0.41
IC	84.58 ± 29.30	0.16 ± 0.36

2.6 Conclusion

We proposed a teleoperation method called IC with (1) comfortable operation with limited information and (2) no loss of task efficiency. The user study results showed that IC offered improved user experience compared to conventional methods, while task efficiency was equivalent in all methods. Further, we attempted to solve two IC issues. Image transformation, which brings the appearance of the v-space closer to the r-space and visual effect that eliminates the gradual separation of v-pos and r-pos over time. The results of the user study showed no difference in comfort or task efficiency with the application of these improvements.

In the design of interfaces for robot operations, IR was proposed as a new alternative to VR, AR, and MR, demonstrating its feasibility.

2.7 Future Work

Toward a true fusion of v-space and r-space: Ideally, IC is to make the boundary between r- and v-space imperceptible to the user. The methods to achieve this concept are explained in Section 2.4. Specifically, these include an image transformation using CycleGAN to bring the appearance of r- and v-spaces closer, and a deceleration and a visual effect to adjust positional differences over time to provide seamless switching between r- and v-space. The effectiveness of these methods was evaluated in Section 2.5.5.3, but no significant differences were observed in any of the evaluation metrics. In other words, these methods still have room for further improvement.

For the image transformation, further improvements are needed to bring the appearance of the r- and v-spaces closer. Figure 2.8 (B) and (C) reveals that, compared to the clarity of the observation images in the r-space, some parts of the v-space appear blurry with indistinct edges between colors. Ideally, it is desirable to reconstruct a v-space that looks exactly the same as the r-space, but there are still many technical challenges to achieve this.

In this research, we focused on transforming the rendered images of the v-space. However, an alternative approach could be to transform the observation images of the r-space. By applying transformations that blur the observation images of the r-space, within a range that does not affect the operator's actions, to make them closer to the rendered images of the v-space, we may achieve visual consistency through mutual

adjustments of both the r- and v-spaces. This approach is worth exploring further.

Improvement for the switching algorithm: In this research, the system was designed to switch from real to virtual when the robot's future trajectory encounters an obstacle. However, there is room for improvement in this criterion for two reasons. Firstly, there are cases where switching is unnecessary for the operator. If the operator is clearly aware of the obstacle and is performing appropriate operation to avoid it, switching to the virtual space might be unnecessary. Secondly, in environments where obstacles are densely arranged, frequent switching could occur. Such frequent switching might introduce new operational inconveniences. Therefore, incorporating elements such as the operator's awareness of obstacles and the characteristics of the obstacles into the switching criteria may enhance the applicability of IC to various environments and situations.

Application to communication delay: We proposed the concept of IC and developed a system for an obstacle avoidance application. We believe it can also be applied to address communication delays using this framework, which involves simulating expected actions in the v-space and switching between r- and v-spaces. For example, in the teleoperation of robots in outdoor environments or indoor environments with insufficient communication infrastructure, it is often challenging to maintain a stable network connection, and there are times when operations must be performed with significant communication delays. To allow operators to perform operations without stress in such environments, one potential method is to simulate the operator's expected actions within the v-space, like a predictive display, and have the system provide feedback images of the v-space to the operator. Additionally, by switching between r- and v-spaces based on the communication conditions, it may be possible to achieve both accurate observation of r-space situations and comfortable operation.

User Study: In this experiment, we compared IC with traditional shared control and direct teleoperation methods. However, there is room for considering comparative experiments with methods combined with xR technologies such as VR/AR/MR. In this experiment, in SCA, in addition to the shared control method, we used a simple alert message to inform the operator that their command was not accepted. However, by applying a xR methods, providing the operator with additional information, such as why the command was not accepted or which obstacle was recognized, could potentially change the operator's impression of the system. Therefore, there is also room

to explore comparing IR with other xR methods to examine differences in operators' impressions of the system.

Additionally, there is room to evaluate IC operations in complex environments. In environments with many obstacles, it is likely that the operator will spend more time operating in the v-space. It is still unclear whether the parameters and methods, such as the deceleration of the virtual robot and the visual effects described in Section 2.4.2, are appropriate for all environments.

3

Illusory Control with Temporal 3D Model

In this chapter, we discuss a method based on Sim-in-Real concept that allows for the instant creation of a temporal 3D model around a robot during operation, using only data obtained in the process, without requiring prior preparation. Additionally, we will describe the results of applying this method in combination with IC to robot teleoperations.

3.1 Introduction

IC is a system that provides a comfortable operation experience using a seamless transition between real and virtual spaces. IC facilitated a safe robot operation without complications in avoiding obstacles because the operators receive feedback images that allows them the satisfaction of feeling that they are operating the robot accord-

ing to their intention by switching to the v-space without obstacles. However, IC was limited because the v-space must be prepared beforehand. For the system to work, a provider or user of an IC system needed to visit the r-space in which the robot moves beforehand, and they needed to sense the environment using a 3D scanner, etc., and post-process, such as adjusting the appearance. Consequently, IC systems could only function in familiar environments. Some applications for a teleoperated robot are difficult to visit beforehand, such as disaster responses. Therefore, the fact that the IC system could only function in known environments severely limited the applicability of IC techniques.

Here, we propose a novel method in which the v-space does not need to be prepared beforehand. Specifically, we leverage the *instant neural graphics primitives (NGP)* [62] technique, a method that is expected to achieve convergence of *neural radiance fields (NeRF)* [21] training in a short time, for the instant construction of v-spaces. We propose *Instant IC*, a teleoperated robot system that adopts a seamless transition between the v-space constructed by instant NGP and the r-space. Furthermore, we propose a depth estimation method to increase the reconstruction accuracy at unfamiliar poses and a scaling method to fit the geometry in the v-space with the r-space. These methods help increase the consistency of the appearance between the v- and r- spaces.

The contributions of this study are as follows.

- The applicability of IC in unknown environments was verified by using instant NGP that can construct a v-space instantly.
- We verified that a prior depth estimation improves reconstruction accuracy in unknown postures.
- We verified that depth scaling based on actual measurements from LiDAR sensors can improve the geometry consistency between r- and v- spaces. Specifically, this is a unique challenge for ICs that use seamless transitions between r- and v- spaces.

The remainder of this chapter is organized as follows. Section 3.2 presents the related work. Section 3.3 describes the background of IC and NeRF. Section 3.4 describes the proposed method, and the experiment is described in Section 3.5. Finally, Section 3.6 concludes this chapter.

3.2 Related Work

Conventional 3D reconstruction techniques include point cloud-, voxel-, and mesh-based methods. The point cloud-based methods cannot represent object surfaces; hence, it is difficult for them to reconstruct a highly realistic 3D space on their own. The voxel-based methods are also memory inefficient and poor at reconstructing accurate geometry. The mesh-based methods can accurately reconstruct geometry and achieve highly realistic 3D space using texture. In fact, we have adopted mesh-based methods [1] described in Chapter A or added mesh-based methods to propose methods [2] described in Chapter 2 for similar v-space and r-space appearances using CycleGAN [19]. However, it is difficult to adapt them to IC, which attempts to circumvent v-space preparation beforehand because preprocessing time is required to reconstruct a 3D model. In IC, considering the need to switch between r-space and v-space in response to obstacles with as little operator discomfort as possible, it is desirable to construct a v-space in a few seconds.

A 2D image-based 3D reconstruction method using neural networks has been proposed [17, 18]. One approach is to estimate depth from images [63, 64, 65]. These images are highly realistic because they are based on actual images and perform geometrical transformations based on posture. Because these methods are image transformations based on a single image, it is difficult to reconstruct a consistent environment using images in several poses, especially for large-scale spatial restoration.

Other approaches include simultaneous localization and mapping (SLAM); in particular, dense SLAM should also be able to achieve a highly realistic reconstruction [66, 67]. These alternative approaches are expected to perform dense and highly realistic 3D reconstruction in real-time. To further improve higher realism, NeRFs have recently garnered significant attention [21]. NeRFs utilize image-posture pairs to train a neural network to perform free-viewpoint rendering. NeRFs are also being studied for the consistent and realistic reconstruction of relatively large outdoor environments [7, 68, 69]. Furthermore, a method combined with SLAM has also been proposed [70, 71]. In addition, a method that considers rendering on mobile devices has been proposed; hence, future developments can be expected to work on inexpensive devices [72]. Recently, a method has been proposed to increase the accuracy of reconstruction in unfamiliar postures by optimizing with prior information on the depth [73]. NeRF with 360° images has also been proposed to enable the comprehensive reconstruction of the environment from a small number of images, independent

of the posture at the time of capture [74]. However, the problem with NeRF was the lack of actual learning and rendering time. Instant NGP is a technique that enables the convergence of learning in a significantly short time [62]. Instant NGP is expected to reconstruct the v-space to accommodate instant IC's requirements. Therefore, the leveraging of instant NGP could be an effective improvement method to make IC preparation unnecessary. In addition, from the aspect of instant NGP, instant IC is an example of effective use of its technology's features.

3.3 Background

3.3.1 Illusory Control

This section describes the teleoperation flow of a mobile robot using the proposed IC.

First, the operator sees the camera image of the r-robot and commences the operation with the r-robot as the control target. The r-robot receives operation commands from the operator and calculates its future trajectory based on these commands. It then determines whether the calculated future trajectory will reach the obstacle or not and, if it does, switches the control target from the r-robot to the v-robot. At this point, the system moves the v-robot to the same position based on the position information of the r-robot, switches the image shown to the operator to the v-space, and then switches the control target to the v-robot. Although the operator operates the v-robot, the r-robot moves autonomously using the position and posture of the v-robot as subgoals. If there are obstacles in the r-space at the v-robot position, the future trajectory of the v-robot is calculated, the cost on the trajectory is obtained, and the point where the cost is below a certain level is set as the subgoal. The robot system periodically acquires the position information of each of the v-robot and r-robot, and when the difference in their respective postures becomes less than a threshold value, it switches the feedback image to the operator from the v-robot to the r-robot and the control target from the v-robot to the r-robot.

Here, we have improved the part of the v-space employed in this teleoperation flow. Specifically, we propose a method that does not require advanced preparation by immediately constructing a v-space using image and posture information obtained while the robot is in operation.

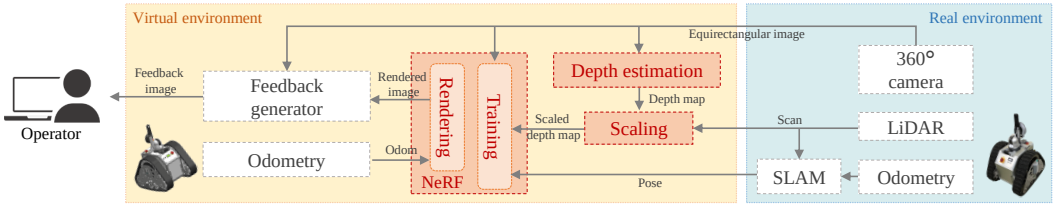


Figure 3.1: Image processing flow in instant IC.

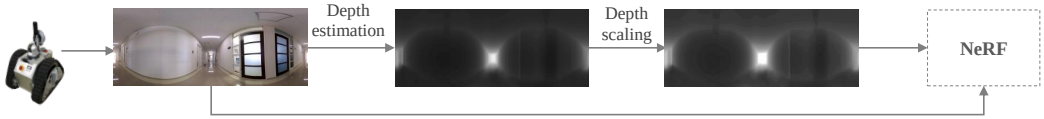


Figure 3.2: Excerpt of the processing part of the image. In this figure, the depth images are adjusted brighter than the actual images to improve legibility.

3.3.2 NeRF

This section describes the issues addressed by the NeRF. NeRF optimizes a function $f(x, d) = (c, \sigma)$ representing a 3D scene, where x, d, c, and σ denote the 3D view position, a view direction, a color, and the density, respectively. Each parameter in NeRF is optimized using a multilayer perceptron (MLP). The rendered color $C(r)$ in some range $n - f$ on the camera ray $r = o + td$ can be defined as

$$C(r) = \int_n^f T(t)\sigma(r(t))c(r(t), d)dt, \tag{3.1}$$

$T(t) = exp(-\int_n^t \sigma(r(s))ds)$ represents the probability that the camera ray terminates at the object surface at t, starting at the neighborhood boundary n. The estimated color C is approximated as

$$\hat{C}(r) = \sum_{i=1}^{N} T_i(1 - exp(-\sigma_i\delta_i))c_i, \tag{3.2}$$

where $T_i = exp(-\sum_{j=1}^{i-1} \sigma_j\delta j)$ and $\delta_i = t_{i+1} - t_i$ represents the distance between two consecutive samples on the ray r.

3.4 Method

The data flow of the proposed method is presented in Figure 3.1. The image data process provided to NeRF is presented in Figure 3.2. The data flow presented here is only the part related to the construction and rendering of the v-space; for the module structure of the entire robot, kindly refer to Chapter 2. Note that this section focuses on the creation of virtual spaces. Other aspects, such as the criteria for switching between r- and v-spaces, are the same as the specifications described in Chapter 2.

3.4.1 Instant Training and Rendering

The process of constructing a v-space using NeRF can be divided into two main parts. The first is a learning process in which r-space image data obtained from the robot under operation are collected and fed to NeRF. The second process involves inputting the posture of the robot on the simulator to NeRF using the learning results and rendering the images seen from that posture.

The NeRF technology employed was instant NGP [62]. This method was adopted because it is expected to achieve learning convergence in a short time, and real-time rendering at approximately 10 fps is feasible by adjusting the rendering quality. Our previous method required advanced preparation of the v-space, thereby resulting in a significant time difference between the current time and the time required to create the v-space. The proposed method can absorb temporal alterations in the environment, except for dynamic obstacles such as humans, because the temporal difference between the r-space and v-space at the time of creation can be maintained from a few seconds to a few dozen seconds.

First, we describe the process of acquiring data from the robot under operation and creating data to be fed to NeRF. Two types of data are acquired from the robot during operation: image and robot-posture information. This information is periodically sent to the computer that constructs the v-space using the communication protocol of the ROS (Robot Operating System). Upon receiving this information, the computer preprocesses the depth estimation and depth scaling described below and then creates and stores the data to be given to NeRF.

The image information used to train NeRF is a 360° image transformed into a total of six perspective images at 90° horizontally and 90° vertically. By employing 360° images, the v-space can be constructed such that the robot's field of view is not limited

by its orientation during data acquisition. When the robot moves a certain distance, the training images are replaced with the latest ones. The posture information used to train NeRF is the robot's posture on the map frame. Specifically, a map frame was constructed while performing SLAM using LiDAR and the robot's wheel odometry, and the posture information for this map frame was adopted. The Gmapping algorithm was employed for SLAM.

Next, the rendering process was simultaneously conducted with instant NGP training. The input was the posture information obtained from the robot on the simulator operated by the operator. The operator was presented with the results rendered by instant NGP as feedback during the operation of the v-robot.

3.4.2 Prior Depth Estimation

If training and rendering are performed by simply providing NeRF with the aforementioned image and posture information, the rendering accuracy for unknown postures will be extremely low. Because the concept of instant IC is to circumvent advance preparation, image information for postures unfamiliar to the robot cannot be obtained beforehand. Therefore, images and postures obtained from the current robot can be employed; however, images and postures at future assumed positions cannot be adopted.

To address this problem, DS-NeRF [73] presented an approach that can render good quality images from a small number of images by providing prior information on the depth. Based on this approach, we attempted to provide prior information on depth. Specifically, NeRF datasets were provided images with pre-estimated depths using the SliceNet [65] algorithm, which can perform direct depth estimation on 360° images (equirectangular). SliceNet was selected because it can achieve depth estimation for 360° images and exhibits the fastest depth estimation time among any of them. Because SliceNet is a supervised learning technique, it is necessary to construct and train a dataset in an environment where depth information is available beforehand. However, this method adopts only the pre-trained model published by the authors of SliceNet, and no additional training, such as fine-tuning, was performed in the current experimental environment.

3.4.3 Depth Scaling

Depth scaling was conducted on the pre-estimated depth images using the actual LiDAR measurements. Because the depth pre-estimation employed with SliceNet is based on a pre-trained model, adopting it as it is in the actual operating environment will trigger a discrepancy in the scale of depth. Because IC is required to switch seamlessly between r-space and v-space, any discrepancy between the scale of the actual environment and that of the model will cause feedback as if the posture is significantly off when the image is switched between r-space and v-space. Therefore, we scaled the pre-estimated depth image using the measured values from a range sensor such as LiDAR. Specifically, we scaled the depth image so that the maximum value of the point cloud information obtained from the range sensor matches the maximum value of the depth image.

The scaled depth D_{ic} is expressed as

$$D_{ic} = \frac{d_{max_l}}{d_{max_s}} D_s \tag{3.3}$$

where d_{max_l} and d_{max_s} represent the max depth values from the range sensor scan and depth map D_s estimated by SliceNet, respectively. Note that d_{max_l} and d_{max_s} denote the maximum values at the 99.5% confidence interval accounting for outliers.

This is expected to reduce the posture shift when the image switches between r-space and v-space.

3.5 Experiment

Instant construction and teleoperation experiments of a v-space with an instant IC were conducted using an actual mobile robot. The mobile robot was a crawler robot manufactured by Ricoh. Ricoh Theta Z1 and Velodyne LiDAR VLP-16 were utilized as sensing devices mounted on the crawler robot. A desktop computer with an Intel Core i9 CPU and NVIDIA RTX 4090 GPU was employed for the v-space construction and teleoperation client.

We aimed to answer the following six questions:

- Can instant IC ensure equivalent reconstruction accuracy when compared to conventional IC, which requires pre-preparation of the v-space?

- How much pre-preparation time is required to build a v-space using instant IC?
- Do the prior depth estimation and depth scaling contribute to ensuring the appearance consistency between v-space and r-space?
- Can instant IC allow the teleoperation of a mobile robot using a transition between v-space and r-space?
- Are there any differences in impressions of the system when compared to instant IC and conventional IC?
- Can instant IC ensure reconstruction accuracy in multiple indoor environments?

3.5.1 Reconstruction Accuracy

In this experiment, we compared the reconstruction accuracy between conventional IC [2] described in Chapter 2 and instant IC. Conventional IC requires a v-space to be constructed beforehand using a 3D scanner and the CycleGAN to adjust the appearance. Instant IC is the proposed method.

For the instant IC, a v-space was constructed using the proposed method based on 360° images obtained from the robot. Only one image obtained at the robot's initial position and the robot's posture at that time were used for the reconstruction using instant IC. We evaluated the degree to which the reconstruction accuracy transitions when the robot moves away from the initial position where the latest image was acquired.

Figure 3.3 illustrates the transitions of the peak signal-to-noise ratio (PSNR) and structural index similarity (SSIM) when the robot moved straight ahead from its initial position. Figure 3.4 illustrates the transitions of PSNR and SSIM when the robot curves forwarding in the corridor in Figure 3.9. Each movement trajectory is presented on the left side of Figures 3.3 and 3.4. The qualitative results of the rendering comparing IC and instant IC are also presented in Figure 3.5.

Focusing on SSIM, it infers that instant IC scores higher than conventional IC in the interval of approximately 2 m from the initial position in the straight and approximately 1–2 m for the curve.

In other words, for a movement range of approximately 2 m, IC can be applied with high reconstruction accuracy by constructing a v-space based on one-shot images using instant IC. In addition, the curving case demonstrates that the v-space construction

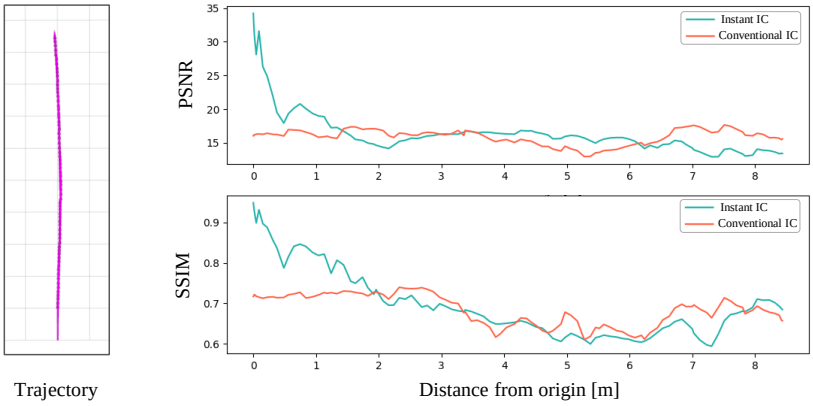


Figure 3.3: Evaluation results of the PSNR and SSIM in the straight trajectory. Instant IC is the proposed method. Conventional IC is a method that requires pre-preparation of the v-space.

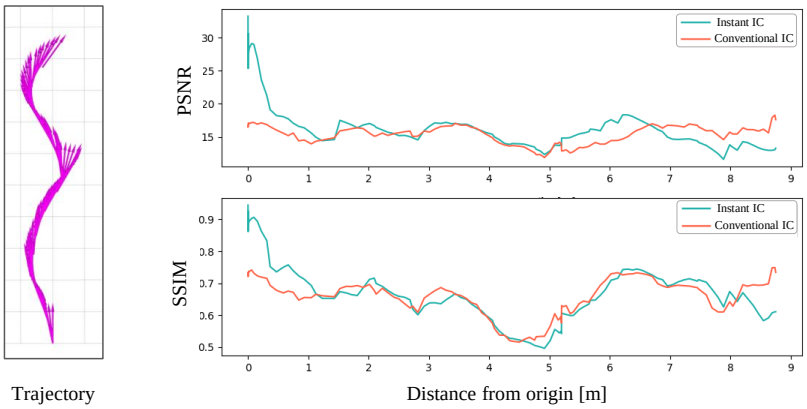


Figure 3.4: Evaluation results of the PSNR and SSIM in the curved trajectory.

with 360° images allows the reconstruction accuracy to be maintained even if there is movement in the turning direction.

This experiment was a v-space construction using only images obtained in the initial posture of the robot to understand the capabilities of instant IC. In an actual teleoperation system with IC, images are acquired, and the v-space is reconstructed every time the robot moves a certain distance; hence, the reconstruction accuracy does not continue to decrease as the robot moves, as illustrated in Figures 3.3 and 3.4.

Figure 3.5: Qualitative results of the reconstruction accuracy. In conventional IC [2], the v-space is constructed by 3D scanner (FARO Focus3D [8]). ①–④ show time-series changes, with the younger numbers indicating earlier times.

3.5.2 Training Time

The time required to learn NeRF was evaluated using the proposed method. Figure 3.6 presents the transition of PSNR and SSIM when the v-space construction using the proposed method commenced with a single 360° image of the robot in its initial posture and drawn in its initial posture.

For both PSNR and SSIM, the scores converge at approximately 3–4 s after learning commences. In other words, if a learning time of approximately 3–4 s is provided beforehand, a v-space with high reconstruction accuracy can be fed back to the operator. Given the teleoperation by IC, the operator will operate the system while seeing images in r-space until they encounter an obstacle; hence, it is hypothesized that a situation may not emerge so much where the operator is made to wait for the v-space to be constructed. Alternatively, the system design may address waiting time by refining the data sampling points or, if computational resources allow, training multiple models.

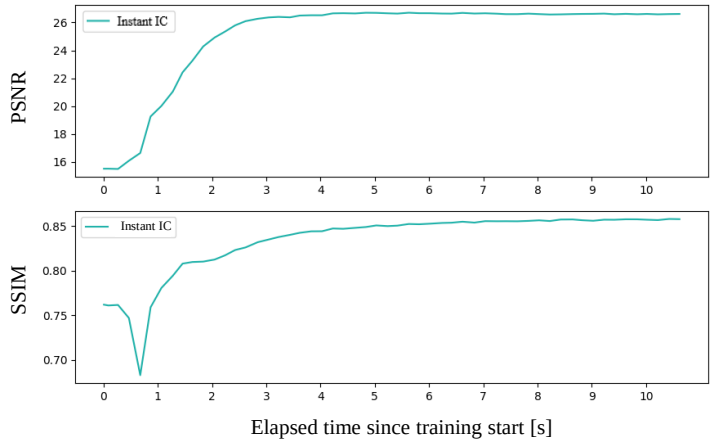


Figure 3.6: Evaluation results of the training time of NeRF.

3.5.3 Ablation Study

The ablation experiment was conducted to evaluate the effects of a prior depth estimation (Ours (-depth)) and depth scaling (Ours (-scale)). Figure 3.7 illustrates the evolution of PSNR and SSIM when moving straight down the corridor from the initial position where the image that functions as the training data was acquired. It appears that Ours (-scale) has the best score in all intervals, although there are a few fluctuations in the scores. However, when observing the rendering results, it appears that

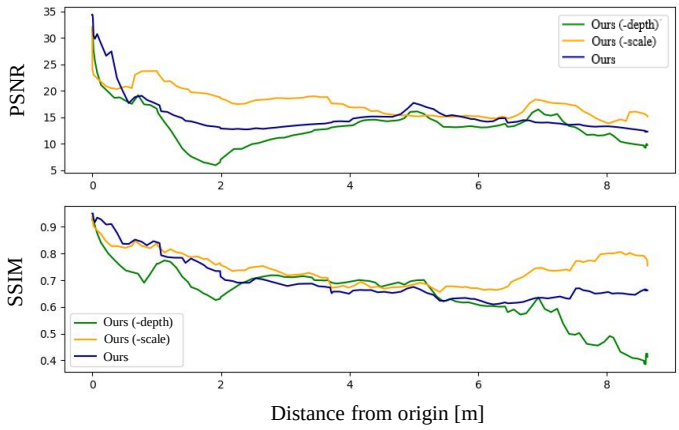


Figure 3.7: Evaluation results of the ablation study in the straight trajectory. Ours (-depth) is the proposed method without the prior depth estimation. Ours (-scale) is the proposed method without the depth scaling. Ours is the proposed method with the prior depth estimation and the depth scaling.

without scaling, the actual geometry of the environment is not correctly represented. Figure 3.8 qualitatively presents the results of rendering via the proposed method using the posture from which the ground truth image was obtained as input. Ours (-depth) has already passed through the mobile environment in ③ and ④. Furthermore, Ours (-scale) has roughly reached the end of the mobile environment at ④. Although there is room for further improvement in the reconstruction accuracy, compared to the above two, Ours appears to be able to reflect geometric information in r-space.

3.5.4 Teleoperation

Figure 3.9 presents the results of a comprehensive teleoperation experiment using instant IC. A white obstacle, which is difficult to recognize from the image, is placed in front of the robot in its initial position. When the robot approaches an obstacle, the feedback image to the operator switches to that of the v-space, and the operator's operation target switches to the v-robot (Figure 3.9 ②). In Figure 3.9 ②-④, the operator operates the v-robot while the robot avoids obstacles by autonomous movements while seeing the images in the v-space. When the difference in posture between the v-robot and r-robot falls below a certain value, the feedback image to the operator switches to that of the r-space, and the robot operated by the operator switches to the r-robot (Figure 3.9 ⑤). The aforementioned operation flow is the same as that for conven-

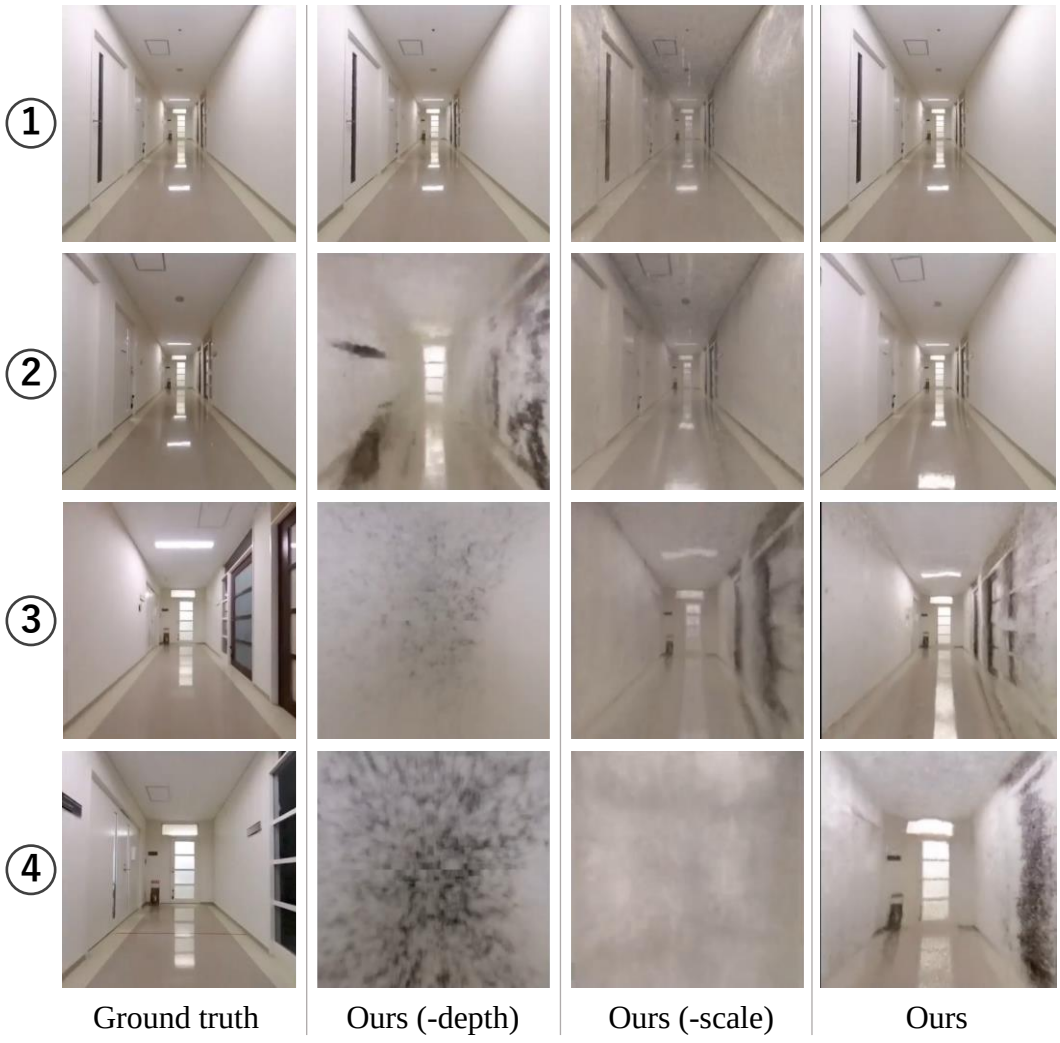


Figure 3.8: Qualitative results obtained from the ablation study. ①–④ show time-series changes, with the younger numbers indicating earlier times.

tional ICs; however, with instant IC, it was verified that this flow could be achieved without preparing a 3D model beforehand. In particular, some roughness emerged in the reconstruction accuracy of areas that were not obtained from the image data, such as the door visible on the right side of the corridor. Improving the reality of these areas is an issue for the future.

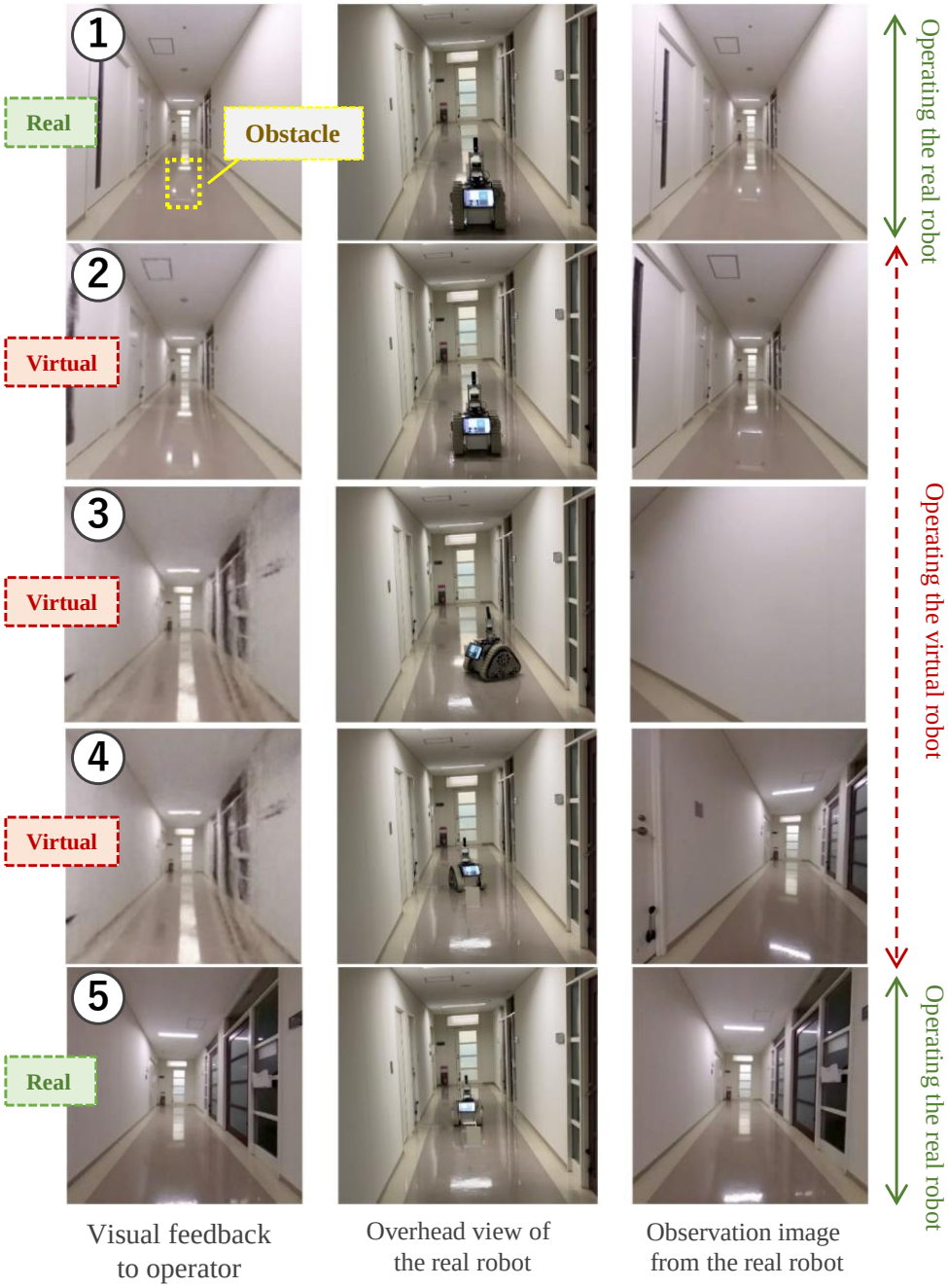


Figure 3.9: Experimental results of the teleoperation by instant IC. ①–⑤ show time-series changes, with the younger numbers indicating earlier times.

3.5.5 User Study

We conducted a user study to verify whether there were differences in impressions of the system operating the robot between conventional and instant IC. The five participants were university researchers specializing in robotics. They participated in the study regardless of their remote control skills and were not compensated. The experiments were approved by the Ethics Review Committee of the Graduate School of Information Science and Electrical Engineering, Kyushu University.

We employed qualitative semi-structured interviews to assess the impressions of the system in terms of realism and usability. This approach was chosen because concerns regarding realism and ease of use differ among individuals, making it challenging to identify these aspects solely through standardized multiple-choice questions. Participants were instructed to remotely control the robot in the environment, as illustrated in Figure 3.9, navigating around two obstacles using the controller while viewing images displayed on the desktop application via both conventional IC and instant IC. The visual effect on V-bot deceleration, as described in Chapter 2, was not implemented in order to evaluate the pure impression of the v-space appearance. To create variations in the timing of the transition to v-space, obstacles were placed in different positions for each participant. This allowed for diverse movement distances from the training image acquisition position and the elapsed time since the training's start, thus providing variations in reconstruction accuracy when transitioning to v-space during the instant IC experiment. Participants were required to experience at least one set of transitions between r-space and v-space in each trial. The experimental order of conventional IC and instant IC was alternated for each participant. During the trial, they were not informed whether the conventional or proposed method was being used. After allowing participants to freely experience teleoperation using both conventional IC and instant IC for approximately 2-3 minutes each, they were asked to respond to the following questions:

- **Appearance**: Which did you find the more realistic appearance, conventional IC or instant IC?
- **Usability**: Which did you find easier to operate, conventional IC or instant IC?

In Chapter 3, no usability improvements were made, but usability questions were included to verify whether differences in appearance affected the ease of operation.

Using the aforementioned questions as a starting point, the interviews proceeded by inquiring why participants felt the way they did. The user study's results showed that for appearance, instant IC received three votes while conventional IC garnered two. In terms of usability, instant IC and conventional IC each received two votes, with one participant considering them equivalent. Below are some comments gathered from the interviews:

- Conventional IC appeared to exhibit a different coloring than the r-space, whereas the instant IC appeared more realistic.
- In the instant IC, the large noise in the image was observed, whereas in the conventional IC, the boundaries of doors and walls could be clearly identified. Consequently, it was easier to recognize the robot's position within the image using the conventional IC.
- Both conventional IC and instant IC have an appearance that is easy to operate.
- In conventional IC, the obstacles present in r-space disappeared in v-space, while in instant IC, the obstacles were carried over from r-space to v-space, providing a sense of consistency.

In instant IC, some participants appeared to feel that the appearance was incomplete. Examples of transitions and less accurate transitions with relatively precise v-space reconstructions are shown in Figure 3.10. Others believed that instant IC seemed more realistic when the operation in v-space commenced at a location close to where the training data were collected. As demonstrated in the learning time experiment, instant IC required a training time of 3-4 seconds. Ideally, learning should converge at the point of transition to v-space, so the design acquires data at fixed movement intervals; however, depending on the timing of the transition to v-space, it may not provide sufficient realism. Moreover, even with adequate training time, when the operation began in v-space at a distance from where the training data were obtained, participants seemed to perceive degradation, such as blurring, from the r-space image.

To address the issue of transitioning from real to virtual before the learning has sufficiently converged, a method to determine convergence based on similarity comparisons with observed images could be implemented. For example, training multiple models with staggered sampling timings of the observed images, and then switching to the model that has converged, is a potential solution.

Just before
the transition to v-sp

Just after
the transition to v-sp

Better case

Just before
the transition to v-sp

Just after
the transition to v-sp

Worse case

Figure 3.10: Examples of transitions and not-so-accurate transitions with relatively accurate reconstruction of v-space in the user study.

Instant IC can accommodate changes in the environment, such as the emergence of obstacles, because the virtual space was created only several seconds prior. In this regard, some participants felt that instant IC operated more consistently. Although obstacles appear visually in the virtual space, they do not have collision detection. If the operator does not notice the presence of an obstacle, they can continue to operate as if the obstacle does not exist. Adapting to environmental changes in a virtual space is an essential aspect of IC. As previously mentioned, as long as realism can be enhanced, instant IC can offer a more consistent operation.

3.5.6 Qualitative Results in Multiple Indoor Environments

In this experiment, the reconstruction was evaluated in various indoor settings. Images were captured at the robot's initial position and utilized for training. After sufficient training, the posture reconstruction was then qualitatively assessed as the robot navigated through an indoor environment. The results of the experiments conducted in two indoor environments are displayed in Figure 3.11.

In relatively small and simple indoor environments, such as corridors, the proposed method can successfully replicate real spaces. However, in some cases, it does not perform well in open indoor environments. Depth estimation and scaling struggle in settings where windows are closely spaced, as depicted in the left column of Figure 3.11, and in complex environments with no windows but featuring intersecting corridors, as seen in the right column of Figure 3.11. Enhancing estimation accuracy to ensure that the method functions correctly in all indoor environments without prior preparation remains a future challenge.

In the depth scaling method used in Instant IC, a constant factor was applied to match the distances observed by LiDAR with the distances in the depth images estimated by SliceNet, focusing on distant points. However, the maximum depth may not always be observable with LiDAR, especially in outdoor environments. There is room for improvement by considering methods such as using the median or the average of several sampled points.

Figure 3.11: Qualitative results of instant IC in multiple indoor environments. The example on the left is of an indoor environment with a view of the exterior beyond the window. The example on the right is of an intricate environment with no windows but with intersecting corridors. ①-⑤ show time-series changes, with the younger numbers indicating earlier times.

3.6 Conclusion and Future Work

3.6.1 Conclusion

Here, we proposed a novel method, instant IC, to eliminate the need for the advanced preparation of v-space in IC. Specifically, we employed instant NGP, a method that is expected to learn NeRF in real-time, to construct the v-space instantaneously. We also proposed a method to improve the rendering accuracy in unfamiliar postures by performing prior depth estimation on images, including a method to improve the agreement with actual geometry by performing depth scaling to the r-space using measured values from LiDAR. Using these proposed methods, we constructed an instant IC system that switched between the v-space created by instant NGP and images in the r-space, evaluated the reconstruction accuracy and training time, verified its operation using a crawler robot, and conducted the user study.

3.6.2 Future Work

Building a technique to interpolate the appearance of areas with missing data is a future task. Possible approaches to solving this issue include using a generative model, such as inpainting, for interpolation [75, 76], ignoring whether it matches the actual visibility, or planning autonomous movement such that it actively acquires areas where data are insufficient [77].

In this approach, a v-space was created using a single image. However, using an image sequence, methods have been proposed that combine SLAM with NeRF learning [71]. Combining our approach with these methods could enable the construction of v-spaces that include past environmental data.

Additionally, similar to Chapter 2, there is room to verify whether Instant IC can be applied in various environments, such as complex environments with many obstacles or spaces wider than corridors.

4

Autonomous Robot Navigation with Long-Term 3D Model

In this chapter, we focus on vision-based autonomous navigation using machine learning, and consider how to efficiently learn navigation policies based on the concept of Sim-in-Real. We describe a framework that creates a long-term 3D model from data observed by a camera, and learns a movement policy that follows the estimated camera trajectory, demonstrating that autonomous navigation is feasible with a mobile robot.

The proposed method, EBIAN, uses a 3D model created by NeRF from an image sequence captured by a camera as the learning environment for the policy. The 3D model learned by NeRF can vary without depending on fixed camera parameters, allowing the teaching camera to be different from the one actually used in autonomous navigation. Furthermore, it does not depend on the camera mounting position on the robot, therefore, the teaching and testing phases can have different mounting positions. Thus, it is also possible to perform teaching by having users capture the route with ev-

eryday devices, though using a camera mounted on the robot is also acceptable. We introduce an approach that seamlessly creates a v-space based on image data obtained during user operations, without the user's conscious intervention for v-space creation, enabling the teaching and learning of autonomous navigation.

4.1 Introduction

Vision-based robot navigation presents a promising avenue, offering high-dimensional data, diversity in applicable areas, and the advantage of compact, cost-effective camera sensors over traditional LiDAR and GPS-based methods. Machine learning and reinforcement learning (RL)–based visual navigation garners interest for its capacity to enable navigation absent maps and explicit rules [78, 79]. Specifically, vision-based imitation learning (V-IL) has emerged as an efficient training strategy over RL-based approaches, enabling robots to learn policies through expert demonstrations of desired behavior effectively [80, 81, 82].

Nevertheless, the practical application of V-IL in robot navigation faces four challenges: **Problem (Prob.) 1**: Environmental damage may arise from collisions with obstacles or walls as robots gather training data in real settings. In the case of IL, human demonstrators are needed to operate a robot in a navigation environment, while learning with RL is even more so because the agent learns the policy by interacting with the environment. **Prob. 2**: Typical IL approaches require state and action pairs, necessitating human facilitation for bringing the robot to the environment and operation within the navigational environment for demonstration acquisition. **Prob. 3**: These methods impose significant human labor to both operate robots for data collection and physically bring robots to environments. **Prob. 4**: The effectiveness of a trained policy depends on the robot configuration, such as camera parameters and robot dimensions, with alterations in camera specifications or mounting positions necessitating the collection of new demonstrations.

This chapter introduces an imitation learning framework named *Environmental and Behavioral Imitation for Autonomous Navigation (EBIAN)*, utilizing Neural Radiance Fields (NeRF). EBIAN features imitating an environment and expected behavior via NeRF, which optimize radiance fields for a photorealistic image rendering and an associated camera trajectory [83, 84], to address these issues by creating photorealistic simulation environments for V-IL training. The capability of NeRF to produce photo-

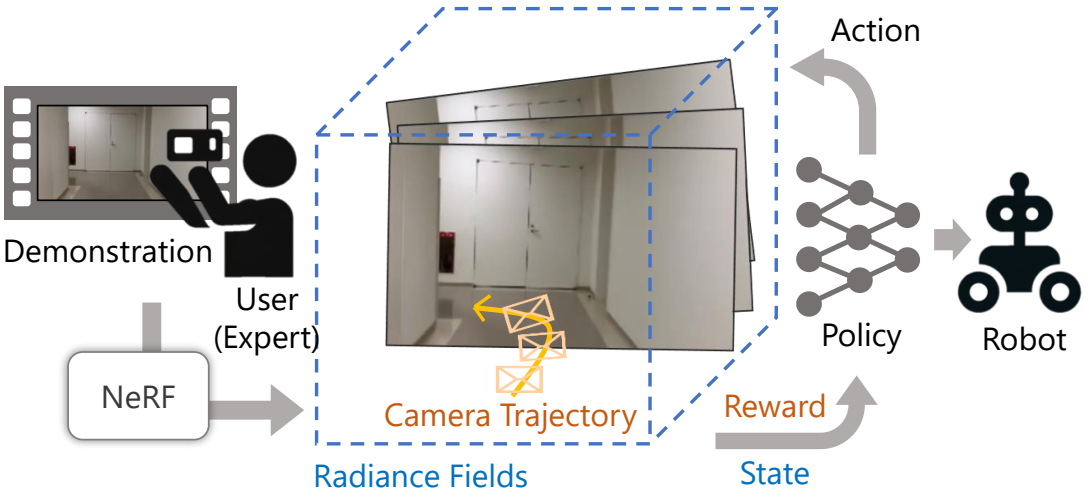


Figure 4.1: The concept of the EBIAN. After users capture images of the navigation course, radiance fields and camera trajectory are estimated by NeRF all at once. They are dealt with as an environment for observing a state and expected behavior for a reward and are leveraged by reinforcement learning agent.

realistic images surpasses traditional 3D reconstruction techniques, such as structure from motion [85], potentially reducing the simulation-to-reality (sim2real) gap [86]. NeRF facilitates free-viewpoint rendering, allowing agents to simulate observations from any chosen pose, thus leveraging NeRF as the simulated environment could eliminate the need for real-world interaction (**Prob. 1**). Moreover, NeRF provides state-action pairs through rendered images and camera poses for navigation training (**Prob. 2**). This approach requires only a one-shot video sequence, captured using simple devices like smartphones, without direct robot operation, obviating the need to bring the actual robot to the environment, and facilitating navigation in environments the robot has not previously encountered (**Prob. 3**). Furthermore, the adaptability of NeRF to camera parameter variations ensures the flexibility of a policy for variation of robot configurations (**Prob. 4**). The concept of our method is shown in Figure 4.1.

Our contributions are as follows:

- We propose a one-shot imitation learning framework for mobile robot navigation, by imitating an environment and expected behavior optimized by NeRF, is capable of sim2real transfer and accommodating diverse intrinsic and extrinsic camera parameters.
- We assess the adaptability of our framework for changes in intrinsic and extrinsic camera parameters—critical aspects of robot configuration.

- We empirically demonstrate that the policy learned by our method successfully works for an actual mobile robot in the real environment.

Note that, at this stage, our method is limited to static environments for autonomous navigation and does not account for the appearance of dynamic objects. Additionally, the focus is on learning policies that follow the path indicated by the expert, and obstacle avoidance on the navigation route is not considered.

The structure of this chapter is outlined as follows: Section 4.2 reviews related work. Section 4.3 details the methodology behind our proposed framework. Section 4.4 describes the experimental setup and findings. Section 4.5 concludes this chapter.

4.2 Related Work

4.2.1 Imitation from Observation

Imitation from observation (IfO) is a methodology that bypasses the need to access an expert's action space, relying solely on optimizing action sequences from observed sequences [87, 88, 89]. IfO enables experts to teach behavior using a camera device not mounted on the robot, such as a smartphone, without operating the physical robot. However, these methods require the robot to interact with environments during the learning phase through RL, posing a risk of environmental damage when interactions occur in real settings. Additionally, utilizing simulation environments face the challenge of high preparation costs. Unlike these methods, our method eliminates the need for real-world interaction and allows users to prepare simulation environments easily by capturing images with everyday devices.

4.2.2 Visual Teach and Repeat

Visual Teach and Repeat (VT&R) represents a conventional strategy for following expert trajectory in mobile robot navigation. One approach has been implemented using visual SLAM techniques like ORB-SLAM [90] and RTAB-MAP [91], offering cost-effective solutions [92]. However, visual SLAM-based approaches often fall short in feature-sparse environments, and their performance heavily relies on the robot configuration. Differences in camera specifications or mounting heights from those used during the SLAM mapping process can impede feature point correspondence extraction and navigation. One of the other approaches, a monocular image-based method

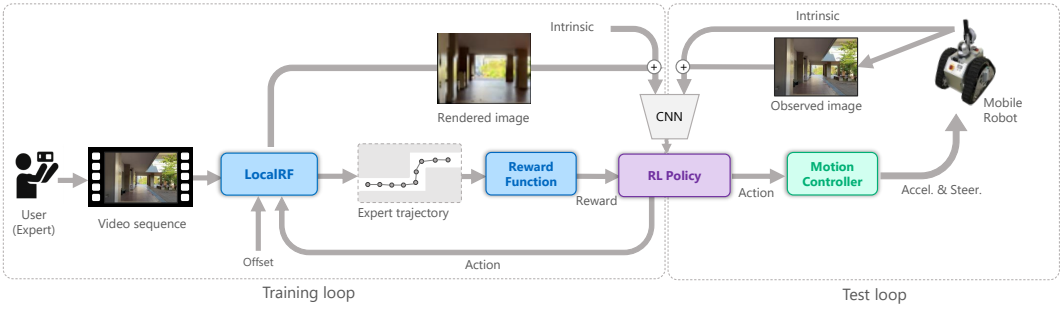


Figure 4.2: Pipeline of the EBIAN.

without SLAM that incorporates misalignment correction [93], still needs accurate robot odometry and bringing the robot to the environment.

Efforts to facilitate VT&R using handheld devices for trajectory teaching have been researched [94, 95]. VOILA [94], for example, achieved autonomous navigation using different camera specifications and mounting heights by learning policies through the correspondence of feature points from image sequences captured by human demonstration. Despite these advancements, such methods still require the robot to interact with the real environment for policy learning, posing risks of environmental damage.

Our approach eliminates the need for real-world interaction, as the navigation policy is learned entirely within a NeRF environment. The capability of optimization of radiance fields and camera poses by NeRF simplifies demonstration acquisition and simulated environment creation only by recording a video along the intended route using everyday devices like smartphones. This circumvents the issues associated with environmental damage and the preparation costs to make simulation environments.

4.2.3 Sim2Real Transfer

Training within simulated environments offers a viable solution to mitigate environmental damage, obviate the need for physical robot transfer during the learning phase, and enhance adaptability to changes in robot configuration. The primary hurdle in leveraging simulations is bridging the domain gap between simulated and real-world environments, known as the sim2real transfer challenge. Traditional methods for this issue have involved domain transformation techniques [6, 20, 27, 28], randomization [29], and adaptation methods [30]. These solutions typically presuppose the creation of 3D simulation environments through labor-intensive modeling processes, leading to significant preparation costs.

Recent advancements with tools like Luma AI [96] and nerfstudio [97] have simplified the generation of photorealistic simulation environments, substantially reducing setup costs. Our approach represents the first effort to take full advantage of the great ease of creating the NeRF environment and to imitate expert behavior by employing NeRF alongside an optimized camera trajectory. Unlike conventional sim2real methods that need image data from both simulated and real environments, our approach requires only a one-shot video sequence to be used as a demonstration.

4.2.4 Robot Navigation with Neural Radiance Fields

Recent initiatives have employed NeRF for vision-based robot navigation [98, 99, 86]. The work presented in [98] aligns with our aim of imitating expert behavior from a one-shot demonstration, applying NeRF to forecast movements and deformations of rigid objects in robot manipulation tasks. Our approach leverages NeRF to simulate both the environment and expert trajectories for visual navigation tasks. The study in [99] has similarities with our approach in the application of NeRF for visual navigation. However, it presupposes comprehensive environmental sensing via camera, differing from our framework, which integrates NeRF-generated environments and expert trajectories from video sequences for learning. Furthermore, it does not validate navigation performance on actual robots, a critical aspect of our method. Additionally, [86] explores the sim2real transfer potential for policies in bipedal robot tasks, focusing on the application of NeRF for behavior optimization. NeRF environment generation and the definition of expected tasks are separate in the [86], whereas our approach defines the environment generation and task definition in one step.

Our work features being the first to employ NeRF-generated radiance fields and an expert trajectory for navigation simulation, specifically evaluating the generalizability across variations in camera intrinsic and extrinsic parameters and demonstrating real-world navigation.

4.3 METHOD

4.3.1 Overview

This section outlines the pipeline of our proposed method, depicted in Figure 4.2. As a 3D reconstruction and pose optimization method, we leverage local radiance fields

(LRF) [84], which optimizes camera poses and radiance fields progressively and is robust for a front-facing video sequence. We propose the V-IL framework named *EBIAN*, using LRF.

In the training loop, users record a video while navigating the intended path. This video is processed by LRF, which then commences the optimization of radiance fields and camera poses, the latter serving as the expert trajectory for navigation. An RL policy is trained using these radiance fields and the trajectory. To accommodate variations in camera mounting positions and specifications, we apply extrinsic and intrinsic offsets to state representations, using height and field of view (FoV) as examples, respectively.

During the testing loop, a mobile robot observes images through its camera. These images are fed into the RL policy as state inputs, based on which the policy generates an action, navigating the robot.

Note that, at the stage of this method, we focus on static environments without obstacles, and it is assumed that the robot's recognition of obstacles is not considered.

4.3.2 State and Action Representation

We define an observation $O_t \in \mathbb{R}^{H \times W \times 3}$ as an RGB image, formulated as follows:

$$O_t = \begin{cases} LRF(P_t, \omega_h), & \textit{if training} \\ I_t(\omega_h), & \textit{if test}, \end{cases} \tag{4.1}$$

where $LRF(P_t, \omega_h)$ denotes an image rendered by LRF, $P_t \in \mathbb{R}^6$ represents the camera pose within the radiance fields at a time step t, I_t is an image captured by a robot's camera, and ω_h is the horizontal FoV of the camera, as an intrinsic parameter.

State representations are derived through a convolutional neural network (CNN), expressed as:

$$S_t = CNN(O_t, C) \tag{4.2}$$

$C \in \mathbb{R}^{H \times W}$ is an intrinsic condition described in Section 4.3.5. The action space is continuous, characterized by Euclid Distance. An action $A_t \in \mathbb{R}^D$ is sampled from the policy $A_t \sim \pi(\cdot|S_t)$, with D dimensionally depends on $SE(2)$ or $SE(3)$ for 2D and 3D movements, respectively. Specifically, A_t is the Euclid Distance between camera poses from timestep t to $t+1$. For ground vehicles, 2D movement (x, y) is applicable, whereas for quadcopters, 3D movement (x, y, z) is pertinent, with x indicating

forward, y leftward, and z upward directions. Additionally, ϕ, θ and ψ denote pose in roll, pitch and yaw direction, respectively.

Given the scale discrepancy between real environments and radiance fields—attributable to monocular image-based pose estimation not aligning with real-world scales—we employ two manually selected keyframes (a, b), such as the endpoints of a corridor, to calculate the scale as follows:

$$sc = \frac{dist(P^{real}(I_a), P^{real}(I_b))}{dist(P_a^{LRF}, P_b^{LRF})}, \tag{4.3}$$

where $dist(a, b)$ denotes Euclid Distance between a and b.

During testing, actions are scaled accordingly:

$$A_t^{real} = sc \times A_t^{LRF}. \tag{4.4}$$

Here, I_a and I_b are visually chosen by humans, with $dist(P^{real}(I_a), P^{real}(I_b))$ measured by movement distance.

4.3.3 Expert Demonstration Extraction

Expert demonstrations $\mathcal{D} = \{(O_1^e, A_1^e), \ldots, (O_n^e, A_n^e)\}$ are derived from poses $P_k^e, k \in [1 \ldots N]$ optimized via LRF, where the superscript by e denotes data from the expert demonstration. For each pose P_t^e, LRF renders an image, which is captured as an observation $O_t^e = LRF(P_t^e, \omega_h)$. The expert action A_t^e is determined as a movement distance $A_t^e = P_{t+1}^e - P_t^e$.

Given that trajectories based on precise step-by-step poses may be overly sensitive to human gait-induced vibrations, resulting in an undesirably meandering path, we apply sampling to select the next expert pose after surpassing a set stride interval, aiming for a smoother trajectory conducive to autonomous navigation.

It is crucial to adapt the raw LRF-estimated poses to the type of mobile robot. For experiments with a non-holonomic ground vehicle, directly applying the yaw direction from LRF poses may lead to suboptimal trajectories. For instance, using action dimensions (x, y, ψ) and training an RL policy on unmodified LRF trajectories could result in inefficient maneuvers, such as unnecessary turns. To address this, we adjust the yaw direction based on the (x, y) movement from the expert trajectory, calculating $\psi = atan(y, x)$. This correction is specific to non-holonomic robots and may not be

necessary for holonomic ones.

During training, expert actions are applied to actions other than those to be optimized. For a non-holonomic robot at timestep t, the action $A_t = (a_t^x, a_t^y, a_t^{e,z}, a_t^{e,\phi}, a_t^{e,\theta}, atan(a_t^y, a_t^x))$ is added to the previous pose $P_{t-1} = (p_{t-1}^x, p_{t-1}^y, p_{t-1}^z, p_{t-1}^\phi, p_{t-1}^\theta, p_{t-1}^\psi)$ to obtain $P_t = P_{t-1} + A_t$, with $a_t^{e,z}, a_t^{e,\phi}, a_t^{e,\theta}$ derived from the expert trajectory. This also should be adjusted according to the type of mobile robot.

4.3.4 Policy Optimization

The reward function for each agent transition $R(O_t, A_t, O_{t+1})$ is designed to encourage imitation of the expert demonstration. The reward increases as the agent's pose P_{t+1} after taking action A_t closely aligns with the expected expert pose P_{t+1}^e.

To encourage smoothness in the agent's path and reduce abrupt directional changes, we incorporate a temporal smoothness regularization from CAPS [100], alongside a regularization based on expert actions A_{t-1}^e, A_t^e to maintain essential action variations, such as turning corners. The reward function is defined as:

$$R(O_t, A_t, O_{t+1}; \mathcal{D}) = \begin{cases} L_p - \lambda L_{caps} & \textit{if } L_p \geqq \gamma \\ 0 & \textit{otherwise}, \end{cases} \tag{4.5}$$

$$L_p = \frac{1}{1 + dist(P_{t+1}, P_{t+1}^e)}, \tag{4.6}$$

$$L_{caps} = dist(A_{t-1}, A_t) - dist(A_{t-1}^e, A_t^e), \tag{4.7}$$

where $dist(a, b)$ calculates the Euclid Distance between positions a and b, and γ serves as a tolerance threshold for the permissible deviation from the expert pose within an episode, and the episode ends when $L_p < \gamma$ or the agent reaches the goal. $L_{caps} \geqq 0$ is the range of value.

In addition to the optimization of Q-value $Q(S, A)$ using the above reward function, we implement BC-SAC [101] to effectively learn a navigation policy that closely fits expert demonstrations and can adaptively return to the expert trajectory from states outside the training distribution. BC-SAC is characterized by its ability to combine elements of IL and RL within a single policy and combines the soft-actor critic objective with behavioral cloning (BC) loss for the actor's objective function, expressed as

follows:

$$\mathbb{E}_{S,A\sim\pi}[Q(S,A) + \mathcal{H}(\pi(\cdot|S))] + \lambda\mathbb{E}_{S,A\sim\mathcal{D}}[\log\pi(A|S)]. \tag{4.8}$$

4.3.5 Camera Configurations

EBIAN enhances policy flexibility to apply for changes in the robot's camera configurations without necessitating retraining. It is crucial to simulate variations in intrinsic and extrinsic parameters during the training phase with offsets.

Extrinsic: NeRF can simulate the free-viewpoint rendering and the policy can observe rendering results that take into account differences in camera mounting position. One example of taking advantage of this benefit is incorporating a height offset to develop a policy that is invariant to camera height. While the expected expert trajectory remains unaltered, an offset is integrated into the current pose for LRF rendering requests, such that $O_t = LRF(P_t, \omega_h)$, $P_t = (p_t^x, p_t^y, p_t^z + \epsilon, p_t^\phi, p_{t-1}^\theta, p_t^\psi)$, where ϵ is an offset parameter. In our method, a random z-axis offset is applied within a specific range in each episode, designed to operate without height information during testing, thus not included in the state representation.

Intrinsic: For intrinsic adaptability, we employ a camera-conditioned state representation, denoted as $S_t = CNN(O_t, C)$. The intrinsic parameter is provided to the LRF varying per episode in the range of $[\omega_{h,min}, \omega_{h,max}]$. An additional channel for the intrinsic condition $C = \frac{\omega_h - \omega_{h,min}}{\omega_{h,max} - \omega_{h,min}}$ is concatenated with O_t. Since intrinsic differences, like FoV, can influence policy outcomes—potentially causing incorrect actions such as premature or delayed turns at corners—they are incorporated as part of the state.

4.3.6 Implementation details

Navigation Policy: Our experimental setup utilizes stable-baselines 3 [102]. The CNN architecture for feature extraction comprises three convolutional layers with kernel sizes of 8x8, 4x4, and 3x3, and strides of 4, 2, and 1, respectively, followed by an LSTM layer.

Offline data for BC: Prior to the RL training phase, an offline dataset for BC is created. The preparation of this dataset incorporates rendering with offsets and the camera-conditioned state representation similarly.

Image data augmentation: Following NeRF2Real [86], our methodology applies image augmentations to enhance the diversity of the training data, including randomized adjustments to brightness, saturation, and hue. We refrain from employing geometric transformations, such as image rotation, to maintain the integrity of spatial relationships.
Motion controller: During testing, a motion controller is essential for translating output actions by the policy into control commands suitable for the mobile robot's configuration. Our experimental setup employs a motion controller designed for a differential two-wheeled robot, issuing commands for linear and angular velocities to follow a dynamically specified target position (x, y, ψ) within the odometry frame.
Inference interval: The inference interval during testing denotes the duration until the robot achieves a predetermined movement amount as dictated by the action output or reaches a timeout (set to 0.5 seconds for our experiments).

4.4 Experiments

We conducted experiments to reveal questions as follows:

- Can EBIAN offer enhanced robustness to variations in robot configurations? (Q1)
- Is the policy developed through EBIAN effective in navigating a real-world robot? (Q2)

To address Q1, we executed a series of closed-loop simulation experiments. In these, the agent was trained within a single NeRF-generated environment and subsequently deployed in other NeRF environments created from video sequences captured under varied conditions. The specifics of these experiments are detailed in Section 4.4.2. For Q2, we undertook real-world navigation trials using an actual mobile robot, as outlined in Section 4.4.3. These experiments aimed to assess the practical applicability and effectiveness of the EBIAN-trained policy in real-world settings.

4.4.1 Training Setup

For our experiments, we utilized servers equipped with NVIDIA A6000 GPUs to train the navigation policy. The selected image resolution for this process was 96×54. The

Corridor 1 Corridor 2 Courtyard

Figure 4.3: The experimental environments were two indoor corridors and a courtyard in the university. (Only in this figure is some image information partially masked.)

Table 4.1: Dataset for evaluation

No.	Use	Device	Height	FoV
1	Training	iPhone	120 cm	60 deg.
2	Test	iPhone	70 cm	60 deg.
3	Test	iPhone	120 cm	60 deg.
4	Test	iPhone	170 cm	60 deg.

range of extrinsic offsets applied was $[-0.25, 0.25]$ relative to the scale of the radiance fields, explicitly affecting the robot's height direction. The simulation predefined FoV, with this study selecting three variations: 50, 60, and 70 degrees (deg.). The stride between two poses within the expert trajectory was set to 0.25 on the scale of the radiance fields. The numbers of RL training steps were the same at each environment among the methods, ensuring adequate rewards. The numbers of BC epochs was the same among the methods, and models with sufficiently small losses were used.

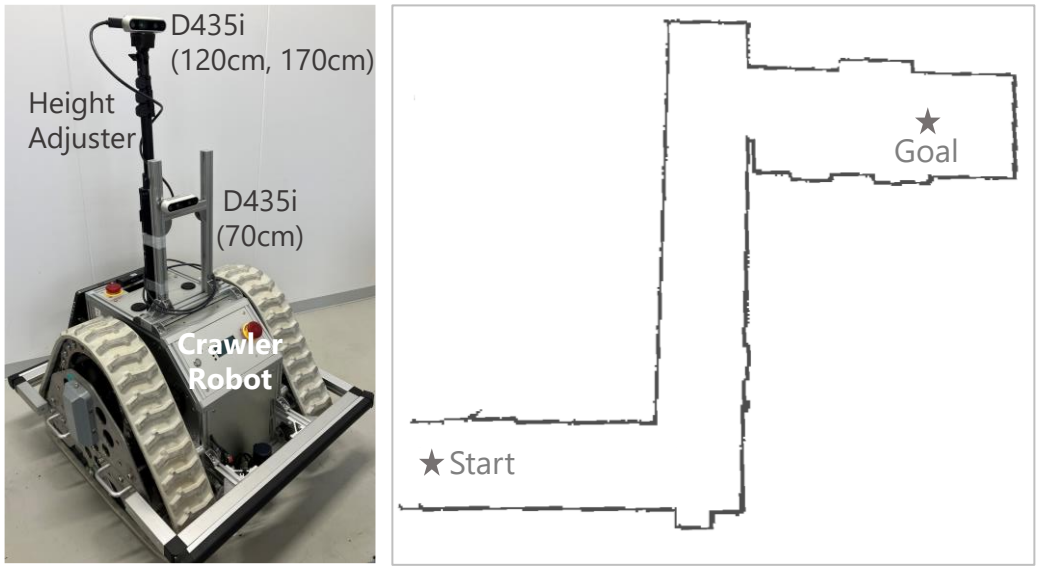


Figure 4.4: (A) The crawler robot with the camera sensor (RealSense D435i). (B) The test course with the actual robot.

4.4.2 Simulation

Setup: The simulation evaluation utilized experimental environments as depicted in Figure 4.3. Data collection involved individuals navigating the course while recording with an iPhone SE3, whose FoV is approximately 60 deg., with the OSMO Mobile SE gymbal to minimize rotation around pitch and roll, at varying heights: 120 centimeters (cm) for training and 70 cm, 120 cm (another set), and 170 cm for testing, as detailed in Table 4.1. This data was processed through LRF to generate radiance fields and camera trajectories, serving as ground truth for simulation evaluation. The objective was to assess the generalizability of the policy across different camera mounting heights and FoVs from the training data.

The policy, trained using radiance fields at a 120 cm camera height, was tested across fields at 70 cm, 120 cm (another set), and 170 cm heights. FoV variations were introduced by adjusting the LRF rendering FoV to 50 deg. and 70 deg. Progress rates, expressed as percentages, were calculated by dividing the step count—up to the point where the distance between expected and current postures exceeded 1.0 (indicating a collision)—by the total step count. A 100% progress rate triggered further comparison based on the sum of positional errors between the agent's and expected poses, with

Table 4.2: Comparison methods

Method	Image	Extrinsic	Intrinsic	SAC
BC (w/ Raw)	Raw	No	No	No
BC (w/ NeRF)	NeRF	No	No	No
EBIAN (-SAC)	NeRF	Yes	Yes	No
EBIAN (-ext)	NeRF	No	Yes	Yes
EBIAN (-int)	NeRF	Yes	No	Yes
EBIAN	NeRF	Yes	Yes	Yes

lower errors indicating superior performance.

Additionally, experiments were also conducted by adding variations to the initial poses within each radiance field. For the existing initial poses, offsets of [-0.1, -0.05, 0.00, 0.05, 0.1] were added in the left-right direction, and offsets of [-0.1, -0.05, 0.00, 0.05, 0.1] were added in the yaw direction, and the same simulation experiments were performed. Combining these offsets resulted in a total of 25 different initial poses. Simulation experiments were conducted for each of these 25 initial poses, and comparisons were made based on the success rate of reaching the goal. The success rate is calculated as the number of successful trials divided by the total number of trials.

Comparison methods are shown in the TABLE 4.2. Behavioral cloning (BC) which has the same network architecture as the actor in EBIAN is set as the baseline. BC assumes expert trajectory acquisition via visual odometry, utilizing LRF-estimated trajectories as labels of training data. BC (w/ Raw) is trained by raw images as the observation, while BC (w/ NeRF) is trained by images rendered by NeRF. EBIAN (-SAC) is set to reveal that RL (SAC) is taking advantage of the ability to collect data in poses not included in the expert trajectory. EBIAN (-ext) is set to reveal the effectiveness of the offset simulation by subtracting extrinsic offsets, and the policy is trained only in 120 cm of capturing height. EBIAN (-int) is without intrinsic offsets, and the policy is trained only in 60 deg. FoV of iPhone.

Results: From the TABLE 4.3, EBIAN demonstrated superior progress rates across all environments and camera height conditions. EBIAN can generalize across all heights in all environments as shown in the 'All' column, although there is partial positional error inferiority at 70 cm in corridor 1. Despite the diminished rendering quality from NeRF with increased extrinsic offsets, i.e., the further away from the estimated camera trajectory, the training using these images proved effective for generalizing across

Table 4.3: Results of evaluation for variation of the camera height. The numbers in parentheses represent the total error compared to the expert's pose and the agent's pose.

	Corridor 1 (Indoor)			Corridor 2 (Indoor)			Courtyard (Outdoor)			All
	60 deg.			60 deg.			60 deg.			
	70 cm	120 cm	170 cm	70 cm	120 cm	170 cm	70 cm	120 cm	170 cm	
BC (w/ Raw)	79	66	18	71	47	44	89	50	43	55
BC (w/ NeRF)	**100 (11.3)**	100 (9.0)	59	40	47	31	100 (7.7)	100 (5.5)	43	71
EBIAN (-SAC)	97	66	42	43	32	31	94	100 (6.0)	100 (3.8)	62
EBIAN (-ext)	100 (21.2)	100 (3.4)	81	43	**87**	90	100 (5.6)	100 (2.2)	100 (6.5)	88
EBIAN	100 (13.7)	**100 (2.9)**	**100 (5.8)**	**100 (4.7)**	**87**	**97**	**100 (1.7)**	**100 (1.0)**	**100 (0.7)**	**98**

Table 4.4: Results of evaluation for variation of camera FoV. The numbers in parentheses represent the total error compared to the expert's pose and the agent's pose.

	Corridor 1 (Indoor)		Corridor 2 (Indoor)		Courtyard (Outdoor)		All
	120 cm		120 cm		120 cm		
	50 deg.	70 deg.	50 deg.	70 deg.	50 deg.	70 deg.	
BC (w/ Raw)	100 (19.9)	47	74	29	50	43	63
BC (w/ NeRF)	75	31	45	47	100 (3.9)	50	53
EBIAN (-SAC)	100 (7.8)	70	45	42	100 (5.9)	100 (8.8)	73
EBIAN (-int)	100 (5.3)	100 (4.8)	**100 (5.1)**	45	**100 (1.1)**	36	87
EBIAN	**100 (3.3)**	**100 (2.6)**	87	**87**	**100 (1.1)**	**100 (1.1)**	**96**

variations in camera height. In particular, the comparison between EBIAN (-ext) and EBIAN suggests that the effect is more pronounced indoors than outdoors, where differences in camera mounting height have a greater impact on the change in appearance.

From the TABLE 4.4, EBIAN progress rates show that the score is high in all but two cases in corridor 2. It can be seen that EBIAN is also effective for the generalization of FoV conditions, however, in the corridor 2 environment, the progress rates did not achieve 100%, possibly due to lower reconstruction quality by NeRF in certain areas, affecting navigation learning accuracy. The cause analysis of failure is explained in Section 4.4.4.

From all scores shown in TABLE 4.3 and 4.4, we found that EBIAN could offer enhanced robustness to variations in robot configurations.

Additionally, from the TABLE 4.5 and 4.6, similar trends were observed in the experiments with success rates when offsets were applied. In all conditions except for the 50 deg. condition in corridor 2, EBIAN achieved the highest success rate. In the corridor 2, even EBIAN had a low success rate. The reasons for these failures will also be discussed in Section 4.4.4.

Table 4.5: Results of evaluation for variation of the camera height with pose offsets

	Corridor 1 (Indoor)			Corridor 2 (Indoor)			Courtyard (Outdoor)		
	60 deg.			60 deg.			60 deg.		
	70 cm	120 cm	170 cm	70 cm	120 cm	170 cm	70 cm	120 cm	170 cm
BC (w/ Raw)	0.00	0.00	0.00	0.08	0.00	0.00	0.12	0.00	0.00
BC (w/ NeRF)	0.44	0.44	0.00	0.00	0.00	0.00	0.76	0.64	0.00
EBIAN (-SAC)	0.24	0.28	0.04	0.08	0.00	0.00	0.64	0.48	0.68
EBIAN (-ext)	0.20	**1.00**	0.08	0.00	0.00	0.00	0.56	**1.00**	0.24
EBIAN	**0.52**	**1.00**	**0.96**	**0.84**	**0.32**	**0.12**	**0.88**	**1.00**	**0.92**

Table 4.6: Results of evaluation for variation of camera FoV with pose offsets

	Corridor 1 (Indoor)		Corridor 2 (Indoor)		Courtyard (Outdoor)	
	120 cm		120 cm		120 cm	
	50 deg.	70 deg.	50 deg.	70 deg.	50 deg.	70 deg.
BC (w/ Raw)	0.16	0.00	0.00	0.08	0.08	0.00
BC (w/ NeRF)	0.32	0.20	0.00	0.00	1.00	0.08
EBIAN (-SAC)	0.52	0.52	0.00	0.00	0.44	0.40
EBIAN (-int)	0.68	**1.00**	**0.68**	0.00	**1.00**	0.00
EBIAN	**1.00**	**1.00**	0.60	**0.24**	**1.00**	**1.00**

4.4.3 Real-World Robot Navigation

Setup: The real-world navigation experiments using EBIAN, utilized a two-wheeled mobile robot equipped with a RealSense D435i camera, as illustrated in Figure 4.4 (A). As in the simulation experiment, the camera mounting height could be changed to 70 cm, 120 cm, and 170 cm conditions. For the FoV adjustment, images from the D435i, whose FoV is approximately 70 deg., were cropped to 50 and 60 deg. FoV and used as observation. The robot's movement was governed by linear velocity and angular velocity, with the motion controller (depicted in Figure 4.2) converting the output of RL policy (x, y) into linear velocity and angular velocity. The experiments were conducted using an NVIDIA Jetson AGX Xavier embedded within the robot. The scale value of the real environment corresponding to the radiance fields was approximately 2.6, and the stride of the actual robot was approximately 0.65 meter. The navigation course, shown in Figure 4.4 (B), corresponds to corridor 2 from Figure 4.3. Each condition underwent three trials, evaluating performance by the average rate of progress, which was determined by the distance progressed over the total distance moved. Trials concluded upon reaching the goal or when a potential collision, approaching within about 10 cm from a wall, was anticipated. Failures attributed to immediate excessive

Table 4.7: Results of evaluation for real-world navigation

	Corridor 2				
	70 cm	120 cm			170 cm
	60 deg.	50 deg.	60 deg.	70 deg.	60 deg.
EBIAN	80 (2/3)	58 (0/3)	88 (1/3)	71 (0/3)	61 (0/3)

turning post-start were considered outliers.

Results: TABLE 4.7 shows the results of real-world navigation at each condition. Numbers in parentheses indicate the number of successes. We found that the policy learned by EBIAN can navigate the actual robot, although the success rates have not reached 100 %.

Figure 4.5 showcases the outcomes of the real-world robot navigation trials with EBIAN, highlighting two successful attempts on the left and an unsuccessful trial on the right. The successful trials demonstrate the robot's capability to navigate straight paths and execute turns at appropriate junctures under various camera height conditions. Notably, in Figure 4.5 (1), the robot adeptly corrects its course from a non-expert state, such as facing a wall, indicative of the effectiveness of free-viewpoint rendering by NeRF in simulating unseen demonstration conditions. However, in Figure 4.5 (2), at a camera mounting height of 120 cm and FoV of 50 deg., the robot failed to execute a timely turn at the second corner. The same behavior was observed at a camera mounting height of 120 cm and FoV of 70 deg. The cause analysis of failure is explained in Section 4.4.4. Successful trials were observed with 70 cm and 120 cm of camera mounting height and FoV of 60 deg. conditions, other conditions were none, which may have been influenced by different types of cameras. This is also explained in Section 4.4.4.

4.4.4 Cause Analysis of Failure

Failed results, noticeable in corridor 2 across both simulation and real-world experiments, are believed to stem from inaccuracies in NeRF reconstruction. Figure 4.6 illustrates a simulation failure in corridor 2, where Figure 4.6 (B) captures a significant camera pose jump mid-turn, marked by a dashed circle. This inaccuracy leads to a notable degradation in image quality when comparing the original (A) and the

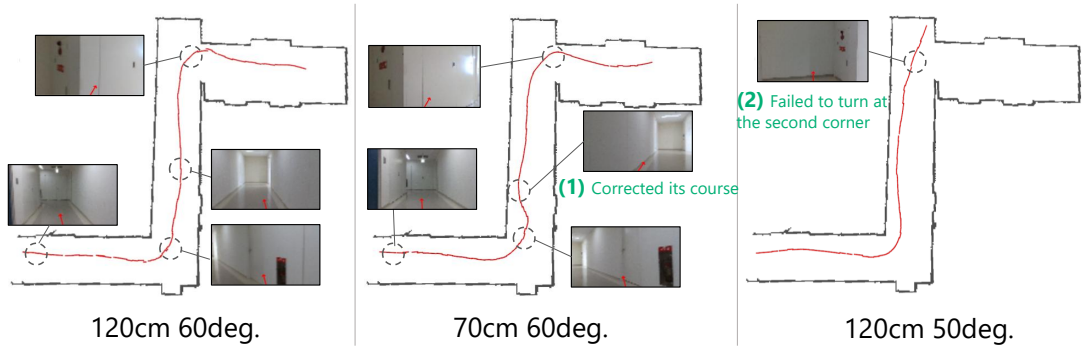


Figure 4.5: Experimental results on actual robot navigation. The map is made by SLAM in advance just to record the trajectory of the movement, the map and pose on it do not affect the autonomous navigation. Red lines indicate the trajectory estimated by AMCL in each trial. The observed image of the vicinity is indicated by the dashed circle, and its action is superimposed with red arrows.

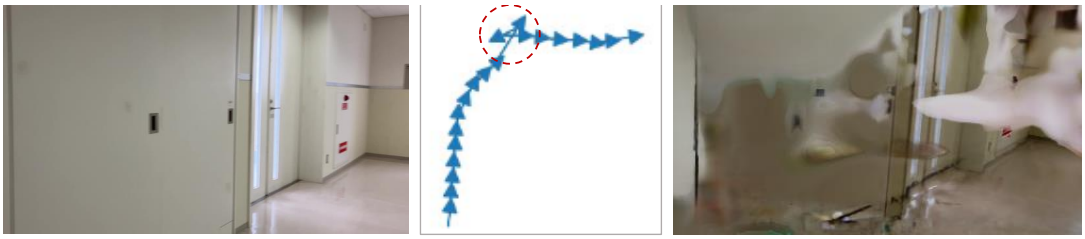
NeRF-rendered image (C), suggesting that navigating within such an inaccurately reconstructed environment could yield unstable outcomes due to the divergence from real-world appearances. Similar failures were observed in real-world navigation tests, as shown in Figure 4.5 (2), where the robot failed to timely navigate a turn. This parallels the simulation issue (Figure 4.6), underscoring the pivotal role of reconstruction precision of NeRF in the successful navigation of EBIAN.

The differences of camera specifications, except for FoV, also contribute to navigation failures. Additional tests by the trained policy in iPhone-generated radiance fields, within radiance fields generated from D435i-captured videos, indicated a performance drop compared to the results of the test in iPhone-generated radiance fields, as seen in TABLE 4.8. The use of D435i, differing from the iPhone, underscores the necessity for policies to accommodate camera-specific differences, including resolution and color, beyond FoV. While EBIAN accounts for intrinsic offsets, further refinement is needed for agents to discern and adapt to camera model variations, with potential solutions like [103] proposed to address these challenges.

Therefore, the enhancement of NeRF is closely linked to the progress in navigation capabilities through EBIAN.

Table 4.8: Results of simulation comparing other types of cameras

	Corridor 2				
	70 cm	120 cm			170 cm
	60 deg.	50 deg.	60 deg.	70 deg.	60 deg.
EBIAN (w/ iPhone)	100	87	87	87	97
EBIAN (w/ D435i)	62	58	66	47	26



(A) Original Image (B) Camera Trajectory (C) Rendered Image

Figure 4.6: (A) The original image. (B) The camera trajectory around the original image. (C) The rendered image is qualitatively low.

4.5 Conclusion and Future Work

4.5.1 Conclusion

This work introduced a novel one-shot imitation learning framework leveraging NeRF to facilitate policy learning without necessitating real-world interaction by the robot. Our method particularly addressed the adaptability of this method to variations in the intrinsic and extrinsic parameters of camera sensors equipped on robots, demonstrating robust performance. Furthermore, we verified the effectiveness of our framework with an actual mobile robot in real-world navigation scenarios. While our approach primarily focused on simple policy learning, generating single-step motion actions from visual observations, there exists the potential for refinement, such as implementing strategies that output multi-step trajectories or actions aimed at minimizing deviations from expected imagery.

4.5.2 Future Work

Our approach demonstrated the capability of an RL agent to learn navigation policies without direct interaction with the real world. Nonetheless, as highlighted in Section 4.3.2, addressing scale inconsistency, potentially through automated scale optimization based on keyframe matching, remains an area for improvement. Additionally, the agent's adaptability to dynamic environmental factors—such as changes in weather, time of day, seasons, and the presence of moving objects—could be enhanced by incorporating techniques like appearance embedding [103] and spatial change representation [104]. Improving the 3D reconstruction quality of NeRF is essential, as the navigation accuracy heavily depends on this aspect, as noted in Section 4.4.4. Thus, advancements in the reconstruction accuracy and rendering quality [105] are anticipated to further refine navigation precision. Future developments should also focus on bridging the gap between the cameras used during the training and testing phases.

The method of defining the expected trajectory for the robot could be expanded. The capturing posture of the expert significantly affects the accuracy of autonomous navigation. Therefore, the expert needs to carefully capture images, considering the images that the robot will observe. If the images are not captured correctly, there is a concern that they will need to be retaken. Additionally, to create a high-precision 3D model that considers all possible camera poses, it is effective to capture a wide range of data by panning the camera left, right, up, and down. However, capturing data in this behavior may result in a discrepancy between the trajectory expected by the expert for the robot and the estimated camera trajectory. Given these factors, besides directly defining the estimated camera trajectory as the expected trajectory, it may be preferable to have options such as sampling efficient camera trajectories from the estimated camera trajectory for autonomous navigation learning.

Although our evaluation was confined to a two-wheeled robot, the underlying policy might apply to other robotic platforms, including holonomic robots and drones with the single policy learned by EBIAN. Extending our evaluation to encompass these additional robot types constitutes a direction for future research. Moreover, we aim to explore the comparative efficacy of our framework against traditional imitation learning and VT&R approaches, which rely on state-action pair data, in various types of simulated environments.

5

Conclusion and Outlook

5.1 Summary

In this research, we contributed to providing intuitiveness in robot operations utilizing Digital Twins (DT). Conventional DT robot systems had issues with the burden on users due to the user interface required to operate the robots and the need for specialized knowledge and significant effort to prepare a virtual space that replicates the real space. These problems limited the number of people who could handle DT robot systems and restricted the fields where DT robot systems could be applied. To solve these two issues and make robot operations using DT more intuitive, we proposed the Ambient Meta World (AMW). AMW is composed of the following two approaches to address the aforementioned problems.

Illusional Reality (IR): Conventional user interfaces for robot operations require users to manage the robot's actions by accurately feeding back the actual state of the robot to the user. In DT robot operations, users must be aware of both the state of the

robot model in the virtual space and the state of the robot in the real space. IR allows the system to take responsibility for the robot's actions, requesting the user to operate only the ideal actions through the virtual space, while the system seamlessly switches without the user noticing. We implemented Illusory Control (IC), which allows the system to switch seamlessly between real and virtual spaces according to the state of the robot and its surrounding environment, enabling users to operate consistently without being particularly aware of the real and virtual spaces. This approach demonstrated the potential to provide a comfortable user experience.

Sim-in-Real: Conventional virtual space creation required pre-sensing and modeling the environment to create 3D models. Additionally, to reduce the visual gap between virtual and real spaces, the Sim-to-Real approach had to be applied. Recently, technologies such as Neural Radiance Fields (NeRF) and Gaussian Splatting have been proposed, allowing the construction of virtual spaces closely resembling the real appearance without needing specialized measuring equipment, using sensors that can be mounted on robots. Therefore, we proposed constructing virtual spaces solely from sensor information obtained during robot operations and examined the potential of applying this to two types of robot operations.

First, we proposed a method that utilizes Instant Neural Graphics Primitives (NGP), a technology based on NeRF, to instantly create a virtual space around a robot from 360-degree images and LiDAR sensor data. We evaluated the quality of the virtual space and implemented Instant Illusory Control (Instant IC) by combining it with IC to assess user experience. This demonstrated the potential to use IC without prior preparation or specialized knowledge.

Second, we proposed a method called Environmental and Behavioral Imitation for Autonomous Navigation (EBIAN), which utilizes Local Radiance Fields (LRF), another technology based on NeRF. This method optimizes camera trajectories and radiance fields used as simulation environments from captured image sequences and learns an autonomous navigation policy that follows the camera trajectory within the radiance fields. We evaluated the performance of autonomous navigation, showing the potential to achieve autonomous robot navigation with minimal operational knowledge and prior preparation.

By proposing and evaluating such AMW approaches, we demonstrated the potential to improve Digital Twin systems into more intuitive ones, minimizing the user's burden and required knowledge.

5.2 Outlook

The findings presented in this research represent initial steps in improving the usability of robot operations through the realization of AMW, yet many challenges remain. The following issues are areas we aim to address in future work:

5.2.1 Issues

5.2.1.1 Illusional Reality

Natural Behavioral Change: To enable users to seamlessly transition between v- and r-spaces, it is necessary to design natural behavioral changes for users.

In IC, users and autonomous agents independently operate the v-robot and the r-robot, respectively. While the v-robot operated by the user moves smoothly without being distracted by obstacles, the r-robot controlled by the agent is focused on obstacle avoidance. To bridge this time gap, we attempted to enhance the sense of speed with simple visual effects as described in Chapter 2, but these visual effects alone did not completely eliminate the sense of discomfort by users. For example, an approach might involve subtly editing the virtual space without a user noticing, such as extending a hallway imperceptibly. By leveraging the human tendency to be less aware of gradual changes, it may be beneficial to adjust the operator's time without causing discomfort. For instance, extending the hallway environment slowly over time could make the operator spend more time, while gradually shortening the hallway environment could reduce the operator's time. Thus, there is still room for further exploration of methods to seamlessly alter human behavior, enabling them to move between virtual and real spaces without any discomfort.

5.2.1.2 Sim-in-Real

Improving the quality of virtual spaces: Both in IC and EBIAN, virtual spaces are constructed based on partial observations. In IC, the virtual space around the robot is built using image data observed at a specific point, but as the observation point is left behind, the appearance of the rendered virtual space degrades. Similarly, in EBIAN, the camera device observes only the image sequence of the desired navigation route, and the virtual space is created based on this image sequence. Naturally, the appearance of the rendering results deteriorates when the camera's pose deviates from

known positions. While traditional DT does not face this issue because they sense the environment comprehensively and create a high-quality virtual space in advance, the virtual spaces created by Sim-in-Real need quality improvements to overcome these limitations.

Adapting to dynamic environmental changes: The virtual spaces created in this research do not account for dynamic environmental changes. In IC, the emergence of dynamic obstacles is technically excluded from consideration. Similarly, the routes used in EBIAN experiments were free from conditions involving dynamic obstacles. Accurately predicting human movements and situations where the robot might get stuck is challenging. There are still many issues to address in order to replicate dynamic elements in a virtual space. However, considering the practical application of the proposed technologies, addressing dynamic obstacles is essential. The system must be capable of recognizing and adapting to changes in the environment to ensure robust and reliable operation.

5.2.2 Utilization of Generative Models

The combination with generative models is a promising approach to address the challenges presented. It is feasible to improve the quality and variety of virtual spaces with a significant workload. However, the direction of this research is to enhance user convenience without increasing their burden. By combining with generative models, we can increase the quality and variety of virtual spaces without adding to user burden. By applying technologies that enable plausible rendering based on past experiences learned by the model and interpolating the virtual spaces created from partial observations, we can expect an improvement in quality. Also, in response to dynamic environmental changes, inserting dynamic obstacles such as humans and moving bodies into the virtual spaces created from partial observations could increase the scene variation. Furthermore, to address the issue of natural behavioral change in IC, it may be possible to present spaces according to the states of the user, v-robot, and r-robot. Thus, we intend to explore extending the use of virtual spaces with minimal user workload and in combination with generative models.

5.3 Conclusion

In recent years, the research and development of generative models have progressed rapidly, unveiling technologies that were unimaginable a few decades ago and overwhelming us with their capabilities. ChatGPT by OpenAI has been continuously updated, significantly enhancing our efficiency in research and development. In fact, the speed of my tasks, such as searching for papers, implementing some ideas, and reading and writing papers, has increased dramatically. However, we believe that the widespread acceptance of ChatGPT is largely due to its interaction design with humans. By embedding GPT-4 technology behind the familiar and widely-used chat interface and tuning the model based on human preference feedback [106], it was seamlessly integrated into everyday use without significantly altering the user experience. A similar example is the introduction of the iPhone, which integrated internet communication interfaces into a phone device, as mentioned in Chapter 1.

The contribution of this research is also significantly related to interface design between users and technologies related to virtual spaces and robotics. With IR, users can seamlessly and intuitively operate the robot's actions using a simple user interface without having to be consciously aware of the virtual or real spaces. Furthermore, with Sim-in-Real, an interface is provided that allows the secondary construction of virtual spaces without requiring special operations to build 3D models, and Sim-in-Real is applicable to temporal and long-term robot operations.

In essence, this dissertation demonstrates an interface design that seamlessly hides the virtual space within our everyday environment, allowing for smooth interactions based on the application's needs. By providing guidelines for interface design that facilitate the seamless integration of the value offered by the combination of robot operations and DT, we hope this dissertation can contribute to the broader acceptance and integration of these technologies into society without causing any discomfort. Not only advanced technology, but also the technology to clearly convey its greatness is necessary.

Appendices

A

Concept of Illusory Control and Verification by Simulator

In Chapter 2, Illusory Control (IC) was a system that represents human operational intentions on a v-space and seamlessly switches between v-robot and r-robot. Humans would operate by switching between r-robot and v-robot. However, when the original IC was initially proposed, the system architecture was different: humans would only operate the v-robot, and the r-robot would autonomously move to follow the v-robot. However, because the r-space environment changes over time and the appearance of dynamic obstacles must also be considered, the system presented in Chapter 2 was introduced. In this chapter, we discuss the original Illusory Control system as proposed and the results of a user study conducted on it.

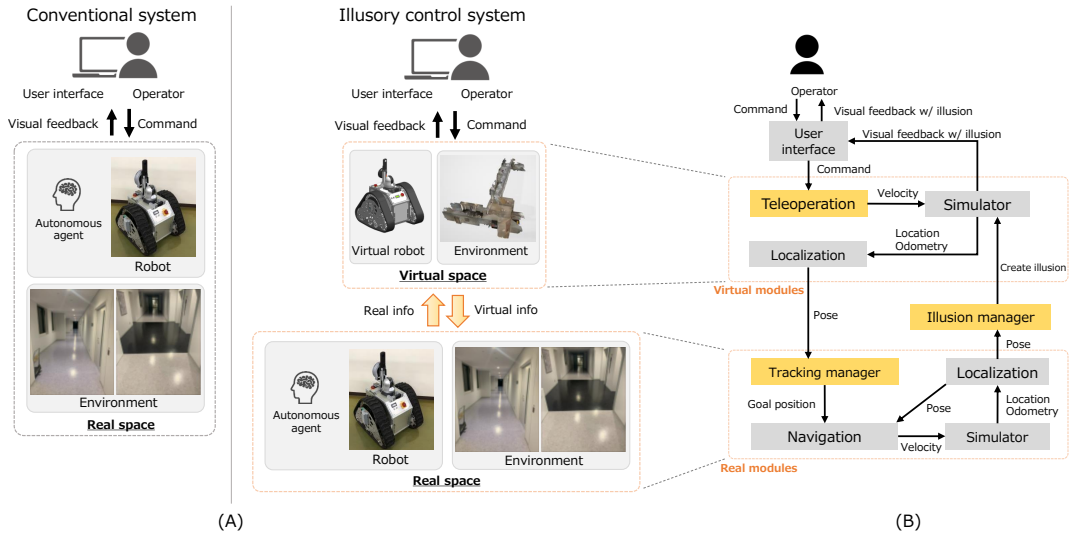


Figure A.1: (A) In conventional systems, an operator and an autonomous agent control the same robot. In the Illusory Control system, the operator controls the robot in the virtual space, and the autonomous robot moves in the real space. The robot moves while interacting with the virtual space and real space. (B) System architecture of Illusory Control. Navigation is performed so that the real-world robot approaches the position of the virtual robot, which is directly controlled by the operator. When the positions of the virtual robot and real-world robot diverge, the Illusion of Time function is activated.

A.1 Illusory Control

The operators may feel stress when they perceive that the teleoperation robot is not moving according to their commands. To address this issue, we devised two methods: (A) not allowing the operator to feel that the robot is not working as expected, and (B) changing the operator commands themselves without the operator noticing.

First, we propose a function named the ”Illusion of Intention” to prevent the operator from feeling that the robot is not moving as expected. Even if the robot automatically avoids obstacles and at times comes to a standstill, if the operator is not informed of this, they will think that the robot is moving smoothly and as expected. That is, the real robot automatically avoids obstacles or stands still, but the operator is not shown the images of the environment in which the robot is located; thus, the operator may feel that the robot is moving according to their intention.

Subsequently, we propose a function named the ”Illusion of Time” as a means to change the operator command itself without the operator noticing. The problem with the Illusion of Intention is that, over time, a difference will exist between the positions

of the human-operated robot and the robot that is actually moving. Therefore, to reduce the difference in positions, the operator commands need to be guided unconsciously to fit the system needs over time. The solution to this problem is to hijack the operator's thoughts by changing the operator commands and the environment that the operator perceives without the operator noticing. This can cause the operator to believe that they are controlling the robot by their own intention, even though they are actually issuing commands that match the intentions of the system (robot). In this study, we implemented the Illusion of Intention and the Illusion of Time based on the above concept. In the following, we describe each implementation in detail.

A.1.1 Illusion of Intention

For the proposed system, we constructed a real space and a virtual space, which was a 3D model of the real space. Moreover, we implemented a set of modules to control the robot in the virtual space and a set of modules to control the real space. The system was implemented using the robot operating system (ROS). The concept of the system and the system architecture are depicted in Figure A.1. First, the operator sends control commands to the robot in the virtual space to cause it to move. While the robot in the virtual space is being moved by the operator, the system estimates the current robot position sequentially. Thereafter, the estimated current position information of the robot in the virtual space is transmitted to a group of modules in the real space.

Subsequently, the modules in the real space perform navigation, using the current position information received from the virtual space as the goal. The navigation is implemented using the Navigation Stack in ROS. Given a goal position, the navigation stack conducts path planning based on Dijkstra's algorithm, and autonomous movement is achieved while avoiding obstacles using the dynamic window approach. While the robot is navigating in the real space, the current robot position in the virtual space changes constantly owing to the continuous teleoperation of the virtual space by the operator. Therefore, based on the amount of robot movement in the virtual space, the new current position is transmitted to the modules in the real space once the robot has moved beyond a certain threshold from the previously estimated position. When the modules in the real space are provided with the position information from the modules in the virtual space, they update the goal point and resume navigation with a new path plan.

This architecture enables the operator to move autonomously in real space safely

while providing the illusion that the robot is operating according to their intention in virtual space.

A.1.2 Illusion of Time

In the teleoperation using the Illusion of Intention described in (A), the problem of the difference between the positions of the robot in the virtual space and real space occurs. One possible solution is to force the operator to stop. However, this may cause strong stress for the operator. Another solution is to guide the operator unconsciously so that this difference becomes smaller over time. The Illusion of Time function is activated based on the current distance of the robot in the virtual space and that in the real space. When the distance between the robots in the virtual space and real space exceeds a certain threshold, the Illusion of Time function is activated. In this study, we devised three methods for realizing the Illusion of Time function and conducted comparison experiments. The methods are described in detail as follows:

Deceleration: This is a method of gradually slowing down the robot's movement in the virtual space according to the distance between the robot in the virtual space and the robot in the real space. Decelerating the movement speed of the robot in the virtual space has the effect of waiting for the robot in the real space to approach the robot in the virtual space.

Blur: In this method, as the distance between the robot in the virtual space and that in the real space increases, it becomes substantially more difficult for the operator to perceive the environment in the virtual space. The operator perceives the virtual environment through the operation user interface (UI) and controls it remotely. When blurring the image of the virtual space and making it difficult for the operator to perceive the space, it becomes difficult for the operator to navigate between obstacles and to perceive the goal ahead. As a result, the operator can expect the effect of decelerating the teleoperation of the robot in the virtual space by their own will and, thus, adjust the time until the distance between the virtual space and real space is reduced.

Obstacle: This method presents an obstacle in front of the robot that is controlled by the operator in the virtual space when the distance between the robot in the virtual space and that in the real space exceeds a certain threshold. When an obstacle is created in front of the operator, the operator must take steps to avoid the obstacle with their own intention. While taking the steps to avoid this obstacle, the effect is waiting for the robot in the real space to approach the position of the robot in the virtual space.

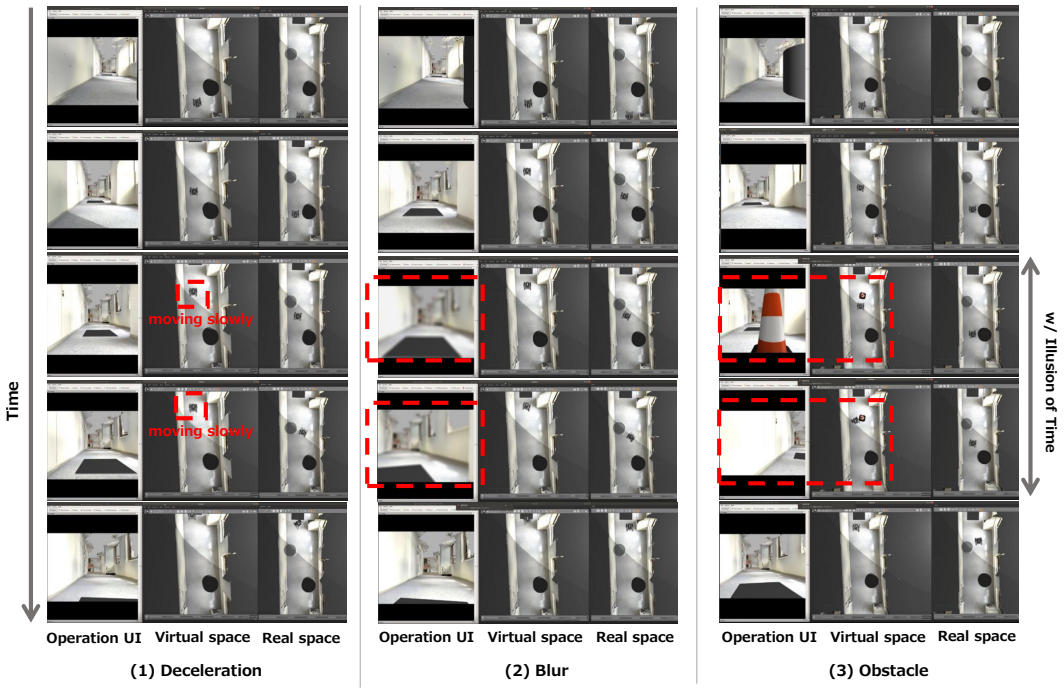


Figure A.2: Illusion of Intention: In (1) to (3), the robot in the real space moves to follow the robot operated in the virtual space. The operator perceives the environment of the virtual space through the operation UI. Illusion of Time: (1) By decelerating the speed, the time required for the real-world robot to catch up with the virtual space can be controlled. (2) The operator operation UI gradually becomes blurred, making it difficult to perceive the environment of the virtual space. (3) An obstacle appeared in front of the robot in the virtual space.

To verify the functional concept of Illusory Control, we constructed both the real space and virtual space on Gazebo to implement a prototype. Figure A.2 presents the activation of each function of the Illusory Control described above.

A.2 Preliminary User Study

We verified the usefulness of the proposed method through a user study in which we compared it with conventional teleoperation methods. We aim to answer the following three questions:

- Can Illusory Control improve task efficiency as well as shared control?
- Can Illusory Control improve system acceptance better than conventional methods?

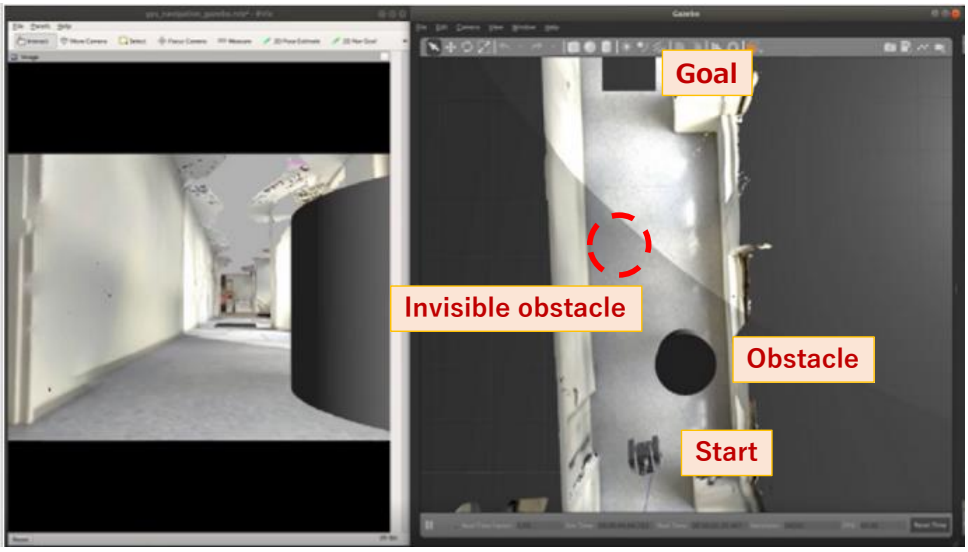


Figure A.3: Experimental environment.

- Which of the three methods using the Illusion of Time is more acceptable?

A total of nine participants (six men and three women) were used. Moreover, we divided the participants into two groups: those who understood the function of Illusory Control (five participants) and those who did not (four participants). This was to verify whether the operation time and impression of the system differed depending on the level of understanding. The tasks to be performed and questionnaire items were the same in both groups. The difference was whether or not the function of Illusory Control was explained to the participant beforehand. In this experiment, the participants were asked to control a mobile robot remotely and to perform a task according to five control methods. The five methods were direct teleoperation, shared control, Illusory Control (deceleration), Illusory Control (blur), and Illusory Control (obstacle). Conventional methods are based on direct teleoperation and shared control. In direct teleoperation, the robot moves according to the directional input from the operator. In shared control, the directional keys of the operator and the local cost map are used as the input, and the local planner of the ROS is used to calculate a path to avoid obstacles and turn the robot in a direction that is free from obstacles.

A.2.1 Experimental Setup

The participants were asked to operate the robot with a PlayStation controller while viewing the operation UI. The robot started from the start point indicated in Figure A.3 and aimed at the goal point while avoiding obstacles. The task was performed once per method. To eliminate order effects, the order in which the five methods were performed was randomized for each participant. Furthermore, the participants could become accustomed to the course as they attempted the five methods. For this reason, we randomly applied two different courses with varying obstacle locations. Moreover, we considered it to be a problem that the robot behavior would change for obstacles that the operator could not perceive without knowing the reason for the change. Therefore, we placed one obstacle that the participant could not see and required them to avoid it. The proposed method, Illusory Control, is based on the assumption that the operator controls the virtual space. Therefore, it is possible to pass through obstacles during the operation of Illusory Control.

An overview of the functions was provided prior to starting the experiment. We did not explain the Illusion of Time function, such as the presentation of obstacles along the way or the difficulty of perception, to the group that did not understand Illusory Control. To evaluate the pure impression of the Illusion of Time function when it was activated, the participants were asked to operate it with no prerequisite knowledge.

A.2.2 Measurements

Both objective and subjective metrics were used to evaluate the usefulness of the proposed method.

We used the operation time from the start to the goal as an objective measure to evaluate the improvement in the operation efficiency. In Illusory Control, there is a virtual space in which operators operate and a real space in which robots move autonomously, but in this experiment, we focused only on the operation time in the virtual space.

As a subjective evaluation, we asked the participants to answer a questionnaire after using each method. This enabled us to evaluate the impressions of the system. First, the System Usability Scale (SUS) [59] was used to evaluate the usability of the system. Thereafter, the subjective impressions of the system were evaluated using four questionnaire items with a five-point Likert scale. Among the four items, the followinn

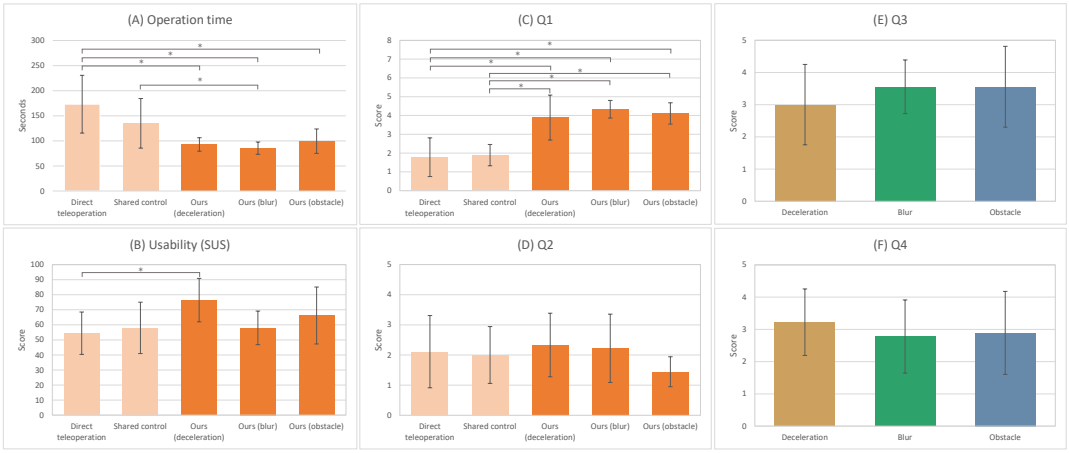


Figure A.4: (A) The proposed method resulted in a shorter operation time compared to the conventional method. (B) A significant difference was observed only when the deceleration method was compared with direct teleoperation. (C) The proposed method improved the acceptance compared to the conventional method. (D) There was no difference in the attention to obstacles between the conventional and proposed methods. (E) and (F) There was no difference between the three Illusion of Time methods.

two items were common to all five methods and were used to evaluate the acceptance of the methods.

- **Q1:** I felt that the robot was working according to my will.
- **Q2:** I took great care not to let the robot collide with any obstacles.

The other two questionnaire items in Table II were used to test the acceptance of the three Illusion of Time methods as follows.

- **Q3:** Before and after the change by Illusion of Time, I felt that I was able to move the robot according to my will.
- **Q4:** After the change by Illusion of Time, I felt it inconvenient to operate the robot.

A.3 Analysis and Results

The experiment was completed by all participants, who were able to reach the goal point from the starting point for all methods. The scores of the experimental results are depicted in Figure A.4. Although we grouped the participants according to whether or not they understood the function of the Illusory Control, no significant difference was exhibited. Therefore, the graph in Figure A.4 summarizes the results for all participants.

The number of participants in the preliminary experiments was small. As a result, some of the measurement results were not normalized or equivariant. Therefore, a one-way analysis of variance was conducted for those with normality and equality of variance, and multiple comparisons were drawn using the Dunnett test as a post hoc test, considering the conventional method as the control group and the three proposed methods as the experimental group. The Kruskal–Wallis test was used for items without normality or equal variance. For items that were compared to the baseline, the Steel test was used as a post hoc test, and for those that were not, the Steel–Dwass test was used to determine whether there was a significant difference.

The operation time results demonstrate that the proposed method was more effective than direct teleoperation in the scenario with obstacles that the operator could not observe from the camera. In the direct teleoperation process, the participants searched for obstacles and attempted to avoid them by trial and error. Furthermore, the proposed method allowed the participant to move through obstacles even when obstacles existed. Therefore, they did not need to be distracted by the obstacles, and the operation time differed.

In terms of usability, a significant difference was observed only when direct teleoperation and Illusory Control (deceleration) were compared. This may be owing to the fact that the participants were less confused than in the other Illusion of Time methods because no change occurred in the appearance of the environment. This phenomenon can also be attributed to the fact that certain participants did not decelerate as much as others. In Illusory Control (obstacle) and Illusory Control (blur), the participants could not understand the reason for the appearance of the obstacles or blurred vision, which suggests that there was no difference from the conventional method.

For Q1, there was a clear difference between the conventional and proposed methods. The reason for this is that in the conventional method, the participant could not operate the robot as desired because the robot stopped moving when it bumped into an

obstacle without understanding the reason or changed its movement direction when it detected an obstacle.

For Q2, there was no difference from the conventional method. In this experiment, we used a simulator for both the virtual space and real space. Therefore, the participants operated the robot in the virtual space using all of the methods. It is desirable to design experiments so that the participant operates the conventional methods in real space. If a difference exists between the operation in the virtual space with the Illusory Control system and that in the real space with the conventional method, the degree to which the participant is concerned about obstacles is expected to differ.

For Q3 and Q4, no difference was observed among the three methods. In all three methods, the participants noticed the change in the environment caused by the Illusion of Time; thus, they were surprised by the sudden change and found it difficult or inconvenient to move the robot as they intended. Therefore, by applying a method of environmental change that the participant does not notice, it is expected that differences will appear in the results. Further improvement of the method in addition to the three Illusion of Time functions described in this paper will be necessary in the future.

A.4 Conclusion

To improve the efficiency of mobile tasks and user acceptance of teleoperated robots, we have proposed a new control method named Illusory Control and conducted experiments on participants using the system. According to the experimental results, a significant difference in the operation time compared to that of direct teleoperation was observed in the scenario with obstacles that the user could not see from the camera. Furthermore, there was a significant difference between the conventional and proposed methods in terms of whether the user could operate the system as desired. In summary, we have demonstrated that Illusory Control offers the potential to increase the efficiency of user operation as much as the shared control and to improve the acceptance of the system more than direct teleoperation and shared control. However, no significant difference was observed in the impression of the Illusion of Time function.

A.5 Limitation

The proposed illusory control in this chapter feeds back virtual space as if it were real to humans. Since this verification was simulation-only, no discrepancy in time between real and virtual spaces was apparent. However, in actual operation, there is a temporal gap between the current time in real space and the time when the virtual space was created. The primary application of illusory control, such as obstacle avoidance, meets functional requirements if virtual space is presented before and after the obstacle. Considering environmental changes over time in real space, presenting the latest situation is desirable. Approaching this temporal discrepancy while considering the desired functionality of Illusory Control is necessary. Therefore, as explained in Chapter 2, IC was updated to a system that takes these issues into account.

B

Illusory Control with Unexpected Situation

In IC, if an unexpected situation occurs, such as the appearance of dynamic obstacles while the user is operating the v-robot, the system is designed to immediately switch back to the r-robot. In normal operation, the system switches back seamlessly from v-state to r-state when the distance between r-pos and v-pos falls below a threshold value. However, the system immediately switches from v-state to r-state in situations requiring a flexible response from the operator, such as the appearance of an unexpected obstacle. As an example, we implemented a function that triggers the detection of a dynamic obstacle, such as a human, and switches back from v-state to r-state. YOLOv5 [107] is used for human detection. The system switches back from v-state to r-state when the number of consecutive human detection frames N_h in the r-robot images exceeds a threshold value. In summary, the v-state and r-state follow the following equation:

$$s(d_{vr}, \theta_{vr}) = \begin{cases} r - state & (d_{vr} < d_{th} \; and \; \theta_{vr} < \theta_{th} \; or \; N_h > N_{th}) \\ v - state & (otherwise), \end{cases} \tag{2.1}$$

where d_{th}, θ_{th} and N_{th} are the threshold parameters.

B.1 Experiment

We verified that the IC's function for detecting an unexpected obstacle and immediately switching the system state from v-state to r-state worked correctly.

The only teleoperation method used in the experiment was instant IC. One obstacle is placed in the robot's movement area beforehand. In the experimental procedure, the operator first starts the operation in r-state. When the robot detects that it is about to collide with an obstacle, it seamlessly switches from the r-state to the v-state. Thereafter, a human (unexpected obstacle) appears in front of the robot at the time of the switch from the r-state to the v-state. Meanwhile, if we can observe that the system state immediately switches from the v-state to the r-state and that the feedback image switches from the v-space to the r-space, then the objective of the experiment is achieved N_{th} was set to 3.

Figure B.1 shows the result of validation during unexpected obstacles. The operator starts the operation in the r-state (Figure B.1 ①). The robot switches from the r-state to the v-state as it approaches an obstacle (②). While the operator operates the v-robot while viewing the v-space, the system detects that a human (unexpected obstacle) has appeared in the r-space (③). The system is forced to switch from the v-state to the r-state because it has been judged that the human has detected for more than a certain threshold period (④).

From the above, we confirmed that the system state immediately switches from v-state to r-state and that the feedback image switches from v-space to r-space.

B.2 Future Work

In the scope of this dissertation, we designed the system to switch immediately to the r-space from the v-space to exclude unexpected situations. However, it is ideally desirable for operations to continue seamlessly. Future improvements could include changing user behavior seamlessly by the system to avoid unexpected situations or replicating unexpected situations within the virtual space.

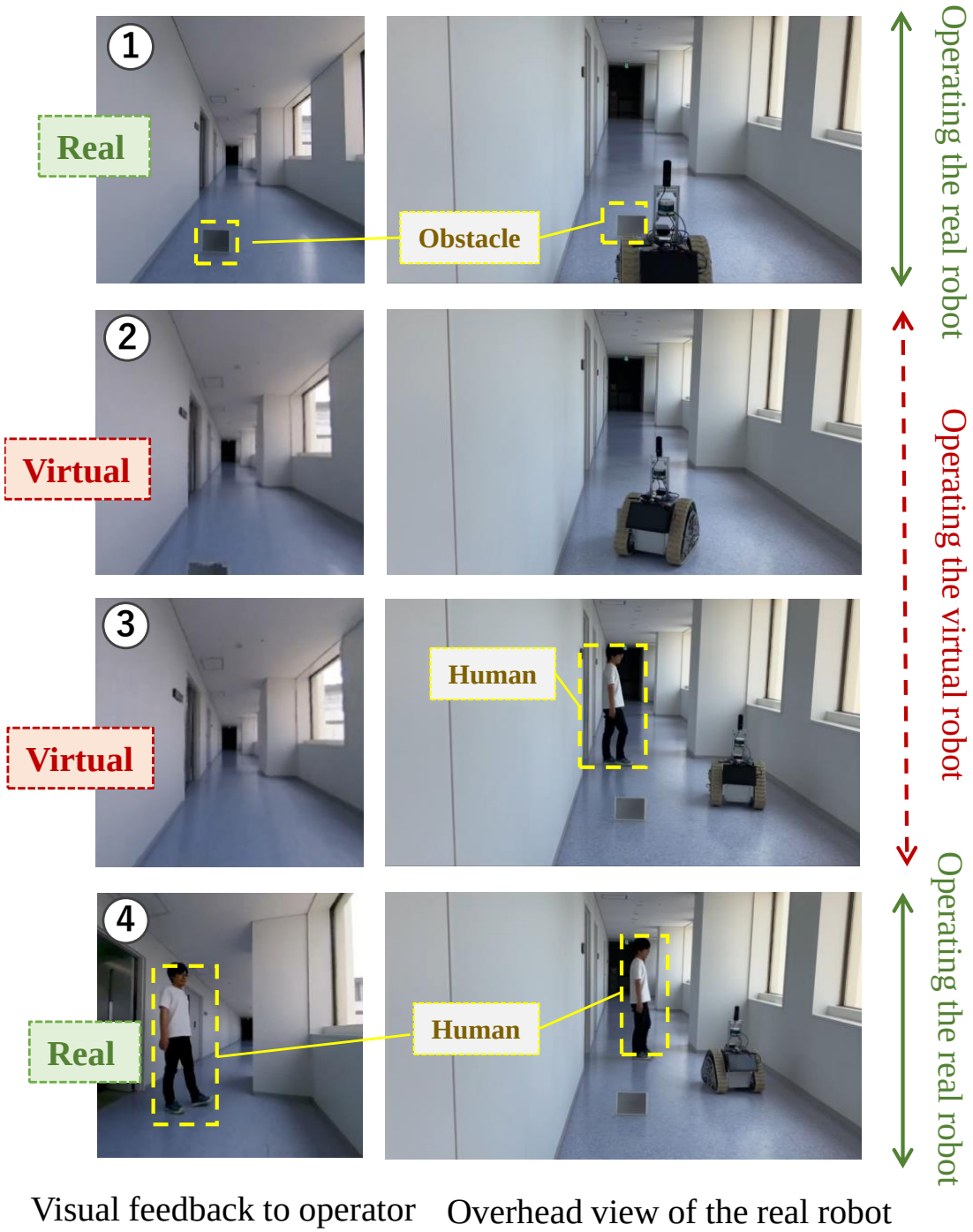


Figure B.1: Verification result of switching back immediately from v-state to r-state when an unexpected obstacle appears

Bibliography

[1] J. Aoki, R. Yamashina, and R. Kurazume, "Teleoperation method by illusion of human intention and time," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pp. 482–487, 2021.

[2] J. Aoki, F. Sasaki, R. Yamashina, and R. Kurazume, "Teleoperation by seamless transitions in real and virtual world environments," *Robotics and Autonomous Systems*, vol. 164, p. 104405, 2023.

[3] J. Aoki, F. Sasaki, R. Yamashina, and R. Kurazume, "Illusory control with instant virtual world environment," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1474–1481, 2023.

[4] J. Aoki, F. Sasaki, K. Matsumoto, R. Yamashina, and R. Kurazume, "Environmental and behavioral imitation for autonomous navigation," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[5] M. Minos-Stensrud, O. H. Haakstad, O. Sakseid, B. Westby, and A. Alcocer, "Towards automated 3d reconstruction in sme factories and digital twin model generation," in *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, pp. 1777–1781, 2018.

[6] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz, and M. Khansari, "Rl-cyclegan: Reinforcement learning aware simulation-to-real," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11154–11163, 2020.

[7] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-NeRF: Scalable large scene neural view synthesis," *arXiv*, 2022.

[8] "FARO Focus3D." `https://www.faro.com/ja-JP/Products/Hardware/Focus-Laser-Scanners`.

[9] C. Office, "Science and technology policy."

[10] C. for Research and D. Strategy, "Research and development trends in digital twins domestically and internationally," *CRDS-FY2021-RR-09*, vol. 2, Mar. 2022.

[11] "Towards next generation digital twin in robotics: Trends, scopes, challenges, and future," *Heliyon*, vol. 9, no. 2, p. e13359, 2023.

[12] B. R. Galarza, P. Ayala, S. Manzano, and M. V. Garcia, "Virtual reality teleoperation system for mobile robot manipulation," *Robotics*, vol. 12, no. 6, 2023.

[13] E. Coronado, S. Itadera, and I. G. Ramirez-Alpizar, "Integrating virtual, mixed, and augmented reality to human–robot interaction applications using game engines: A brief review of accessible software tools and frameworks," *Applied Sciences*, vol. 13, no. 3, 2023.

[14] X. Xu, M. You, H. Zhou, Z. Qian, and B. He, "Robot imitation learning from image-only observation without real-world interaction," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 3, pp. 1234–1244, 2023.

[15] H. Laaki, Y. Miche, and K. Tammi, "Prototyping a digital twin for real time remote control over mobile networks: Application of remote surgery," *IEEE Access*, vol. 7, pp. 20325–20336, 2019.

[16] J. Zhou, Y. Zhou, B. Wang, and J. Zang, "Human–cyber–physical systems (hcpss) in the context of new-generation intelligent manufacturing," *Engineering*, vol. 5, no. 4, pp. 624–636, 2019.

[17] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[18] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[19] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *IEEE International Conference on Computer Vision (ICCV)*, pp. 2242—2251, 2017.

[20] D. Ho, K. Rao, Z. Xu, E. Jang, M. Khansari, and Y. Bai, "Retinagan: An object-aware approach to sim-to-real transfer," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10920–10926, 2021.

[21] 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," in *ECCV*, 2020.

[22] C. S. Tzafestas, "Virtual and mixed reality in telerobotics: A survey," in *Industrial Robotics* (L. K. Huat, ed.), ch. 23, Rijeka: IntechOpen, 2006.

[23] 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*, CHI '22, (New York, NY, USA), Association for Computing Machinery, 2022.

[24] K. Suzuki, S. Wakisaka, and N. Fujii, "Substitutional Reality System: A Novel Experimental Platform for Experiencing Alternative Reality," *Scientific Reports*, vol. 2, p. 459, June 2012.

[25] M. Weiser, "The computer for the 21st century," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, p. 3–11, jul 1999.

[26] E. Aarts and R. Wichert, *Ambient intelligence*, pp. 244–249. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[27] X. Xu, M. You, H. Zhou, Z. Qian, and B. He, "Robot imitation learning from image-only observation without real-world interaction," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 3, pp. 1234–1244, 2023.

[28] A. Bewley, J. Rigley, Y. Liu, J. Hawke, R. Shen, V.-D. Lam, and A. Kendall, "Learning to drive from simulation without real world labels," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 4818–4824, 2019.

[29] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017.

[30] M. Wulfmeier, A. Bewley, and I. Posner, "Addressing appearance change in outdoor robotics with adversarial domain adaptation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1551–1558, 2017.

[31] A. Egashira, Y. Horikawa, T. Hayashi, A. Kawamura, and R. Kurazume, "Near-future perception system: Previewed reality," *ADVANCED ROBOTICS*, vol. 35, no. 1, pp. 19–30, 2021.

[32] S. Johnson, I. Rae, B. Mutlu, and L. Takayama, "Can you see me now? how field of view affects collaboration in robotic telepresence," *ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 2397—-2406, 2015.

[33] H. Tokushige, T. Narumi, S. Ono, Y. Fuwamoto, T. Tanikawa, and M. Hirose, "Trust lengthens decision time on unexpected recommendations in human-agent interaction," *International Conference on Human Agent Interaction (HAI)*, pp. 245–252, 2017.

[34] J. R. Medina, T. Lorenz, and S. Hirche, "Considering human behavior uncertainty and disagreements in human–robot cooperative manipulation," *Trends in Control and Decision-Making for Human–Robot Collaboration Systems*, pp. 207–240, 2017.

[35] S. Tachi, "Telexistence," *Virtual Realities: International Dagstuhl Seminar, Dagstuhl Castle, Germany, June 9-14, 2013, Revised Selected Papers*, pp. 229–259, 2015.

[36] N. Diolaiti and C. Melchiorri, "Teleoperation of a mobile robot through haptic feedback," *IEEE International Workshop HAVE Haptic Virtual Environments and Their*, pp. 67–72, 2002.

[37] F. Okura, Y. Ueda, T. Sato, and N. Yokoya, "Teleoperation of mobile robots by generating augmented free-viewpoint images," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 665–671, 2013.

[38] K.-H. Lee, U. Mehmood, and J.-H. Ryu, "Development of the human interactive autonomy for the shared teleoperation of mobile robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1524–1529, 2016.

[39] S. Gholami, V. R. Garate, E. De Momi, and A. Ajoudani, "A probabilistic shared-control framework for mobile robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11473–11480, 2020.

[40] S. Reddy, S. Levine, and A. D. Dragan, "Shared autonomy via deep reinforcement learning," *Robotics: Science and Systems: online proceedings*, vol. 2018, 2018.

[41] C. Brooks and D. Szafir, "Visualization of intended assistance for acceptance of shared control," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11425–11430, 2020.

[42] Q. Roy, F. Zhang, and D. Vogel, "Automation accuracy is good, but high controllability may be better," *ACM Conference on Human Factors in Computing Systems (CHI)*, no. 520, pp. 1–8, 2019.

[43] C. Ha, J. Yoon, C. Kim, Y. Lee, S. Kwon, and D. Lee, "Teleoperation of a platoon of distributed wheeled mobile robots with predictive display," *Autonomous Robots*, vol. 42, no. 8, pp. 1819—1836, 2018.

[44] H. Dybvik, M. Løland, A. Gerstenberg, K. B. Slåttsveen, and M. Steiner, "A low-cost predictive display for teleoperation: Investigating effects on human performance and workload," *International Journal of Human-Computer Studies*, vol. 145, pp. 1–18, 2021.

[45] Y. Fu, W. Lin, J. Huang, and H. Gao, "Predictive display for teleoperation with virtual reality fusion technology," *Asian Journal of Control*, vol. 23, pp. 2261––2272, 2021.

[46] T. Togias, C. Gkournelos, P. Angelakis, G. Michalos, and S. Makris, "Virtual reality environment for industrial robot control and path design," *Procedia CIRP*, vol. 100, pp. 133–138, 2021.

[47] 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*, vol. 111, pp. 1077—-1097, 2020.

[48] Y. Chen, B. Zhang, J. Zhou, and K. Wang, "Real-time 3d unstructured environment reconstruction utilizing vr and kinect-based immersive teleoperation for agricultural field robots," *Computers and Electronics in Agriculture*, vol. 175, no. 105579, 2020.

[49] S. Jung, P. J. Wisniewski, and C. E. Hughes, "In limbo: The effect of gradual visual transition between real and virtual on virtual body ownership illusion and presence," *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 267–272, 2018.

[50] A. Almutawa, "Effect of smooth transition and hybrid reality on virtual realism: A case of virtual art gallery," *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 3, pp. 231–240, 2021.

[51] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423, 2016.

[52] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, 2017.

[53] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9068–9079, 2018.

[54] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow, "Deep blending for free-viewpoint image-based rendering," *ACM Transactions on Graphics*, vol. 37, no. 257, pp. 1–15, 2018.

[55] A. Ishii, I. Suzuki, S. Sakamoto, K. Kanai, K. Takazawa, H. Doi, and Y. Ochiai, "Optical marionette: Graphical manipulation of human's walking direction," *ACM Symposium on User Interface Software and Technology (UIST)*, pp. 705—-716, 2016.

[56] N. C. Nilsson, T. Peck, G. Bruder, E. Hodgson, S. Serafin, M. Whitton, F. Steinicke, and E. S. Rosenberg, "15 years of research on redirected walking in immersive virtual environments," *IEEE Computer Graphics and Applications*, vol. 38, no. 2, pp. 44–56, 2018.

[57] W. H. Warren, "Self-motion:visual perception and visual control," *Perception of Space and Motion*, pp. 263–325, 1995.

[58] H. Hu, C. Perez, H.-X. Sun, and M. Jagersand, "Performance of predictive display teleoperation under different delays with different degree of freedoms," *International Conference on Information System and Artificial Intelligence (ISAI)*, pp. 380–384, 2016.

[59] J. Brooke, "Sus: A quick and dirty usability scale," *Usability evaluation in industry*, 1996.

[60] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," *Advances in Psychology*, vol. 52, pp. 139–183, 1988.

[61] S. Haga and N. Mizukami, "Japanese version of nasa task load index: Sensitivity of its workload score to difficulty of three different laboratory tasks," *The Japanese journal of ergonomics*, vol. 32, no. 2, pp. 71—-79, 1996.

[62] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics*, vol. 41, no. 4, 2022.

[63] R. Hu, N. Ravi, A. C. Berg, and D. Pathak, "Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image," *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12508–12517, 2021.

[64] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," 2023.

[65] G. Pintore, M. Agus, E. Almansa, J. Schneider, and E. Gobbetti, "SliceNet: deep dense depth estimation from a single indoor panorama using a slice-based representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11536–11545, June 2021.

[66] K. M. Jatavallabhula, G. Iyer, and L. Paull, "∇slam: Dense slam meets automatic differentiation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2130–2137, 2020.

[67] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," *Advances in neural information processing systems*, 2021.

[68] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12922–12931, June 2022.

[69] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, "Urban radiance fields," *CVPR*, 2022.

[70] E. Sucar, S. Liu, J. Ortiz, and A. Davison, "iMAP: Implicit mapping and positioning in real-time," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[71] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," 2022.

[72] Z. Chen, T. Funkhouser, P. Hedman, and A. Tagliasacchi, "Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures," *arXiv preprint arXiv:2208.00277*, 2022.

[73] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised NeRF: Fewer views and faster training for free," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[74] K. Gu, T. Maugey, S. Knorr, and C. Guillemot, "Omni-nerf: Neural radiance field from 360° image captures," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2022.

[75] R. Fridman, A. Abecasis, Y. Kasten, and T. Dekel, "Scenescape: Text-driven consistent scene generation," 2023.

[76] E. R. Chan, K. Nagano, M. A. Chan, A. W. Bergman, J. J. Park, A. Levy, M. Aittala, S. D. Mello, T. Karras, and G. Wetzstein, "GeNVS: Generative novel view synthesis with 3D-aware diffusion models," in *arXiv*, 2023.

[77] H. Zhan, J. Zheng, Y. Xu, I. Reid, and H. Rezatofighi, "Activermap: Radiance field for active mapping and planning," 2022.

[78] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016.

[79] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8248–8254, 2019.

[80] A. Hu, G. Corrado, N. Griffiths, Z. Murez, C. Gurau, H. Yeo, A. Kendall, R. Cipolla, and J. Shotton, "Model-Based Imitation Learning for Urban Driving," Oct. 2022.

[81] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," in *Proceedings of the 1st Annual Conference on Robot Learning* (S. Levine, V. Vanhoucke, and K. Goldberg, eds.), vol. 78 of *Proceedings of Machine Learning Research*, pp. 357–368, PMLR, 13–15 Nov 2017.

[82] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4693–4700, 2018.

[83] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5721–5731, 2021.

[84] A. Meuleman, Y.-L. Liu, C. Gao, J.-B. Huang, C. Kim, M. H. Kim, and J. Kopf, "Progressively optimized local radiance fields for robust view synthesis," in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16539–16548, June 2023.

[85] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[86] A. Byravan, J. Humplik, L. Hasenclever, A. Brussee, F. Nori, T. Haarnoja, B. Moran, S. Bohez, F. Sadeghi, B. Vujatovic, and N. Heess, "Nerf2real: Sim2real transfer of vision-guided bipedal motion skills using neural radiance fields," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9362–9369, 2023.

[87] Y. Liu, A. Gupta, P. Abbeel, and S. Levine, "Imitation from observation: Learning to imitate behaviors from raw video via context translation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1118–1125, 2018.

[88] F. Torabi, G. Warnell, and P. Stone, "Generative adversarial imitation from observation," June 2019.

[89] F. Torabi, G. Warnell, and P. Stone, "Recent advances in imitation learning from observation," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 6325–6331, 2019.

[90] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[91] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[92] M. Mahdavian, K. Yin, and M. Chen, "Robust visual teach and repeat for ugvs using 3d semantic maps," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8590–8597, 2022.

[93] D. Dall'Osto, T. Fischer, and M. Milford, "Fast and Robust Bio-inspired Teach and Repeat Navigation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Prague, Czech Republic), pp. 500–507, IEEE, Sept. 2021.

[94] H. Karnan, G. Warnell, X. Xiao, and P. Stone, "Voila: Visual-observation-only imitation learning for autonomous navigation," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2497–2503, 2022.

[95] M. Fehr, T. Schneider, M. Dymczyk, J. Sturm, and R. Siegwart, "Visual-inertial teach and repeat for aerial inspection," 2018.

[96] "Luma AI." `https://lumalabs.ai/`.

[97] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, "Nerfstudio: A modular framework for neural radiance field development," in *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, 2023.

[98] P. Gesel, N. Sojib, and M. Begum, "Self-supervised visual motor skills via neural radiance fields," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3712–3718, 2023.

[99] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-only robot navigation in a neural radiance world," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.

[100] S. Mysore, B. Mabsout, R. Mancuso, and K. Saenko, "Regularizing action policies for smooth control with reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1810–1816, 2021.

[101] Y. Lu, J. Fu, G. Tucker, X. Pan, E. Bronstein, R. Roelofs, B. Sapp, B. White, A. Faust, S. Whiteson, D. Anguelov, and S. Levine, "Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7553–7560, 2023.

[102] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[103] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7210–7219, June 2021.

[104] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6494–6504, 2021.

[105] R. Wu, B. Mildenhall, P. Henzler, K. Park, R. Gao, D. Watson, P. P. Srinivasan, D. Verbin, J. T. Barron, B. Poole, and A. Holynski, "Reconfusion: 3d reconstruction with diffusion priors," 2023.

[106] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," 2022.

[107] G. Jocher, "Yolov5," `https://github.com/ultralytics/yolov5`, 2020.