QUEEN MARY UNIVERSITY OF LONDON

UNIVERSITY OF GENOA

Exploring Robot Teleoperation in Virtual Reality

presented by

Bukeikhan Omarali

supervised by

Dr Ildar Farkhatdinov
Prof Maurizio Valle

A thesis submitted for the degree of

Joint Doctorate in Interactive Cognitive Environments

Cycle XXXIII

Abstract

This thesis presents research on VR-based robot teleoperation with a focus on remote environment visualisation in virtual reality, the effects of remote environment reconstruction scale in virtual reality on the human-operator's ability to control the robot and human-operator's visual attention patterns when teleoperating a robot from virtual reality. A VR-based robot teleoperation framework was developed, it is compatible with various robotic systems and cameras, allowing for teleoperation and supervised control with any ROS-compatible robot and visualisation of the environment through any ROS-compatible RGB and RGBD cameras. The framework includes mapping, segmentation, tactile exploration, and non-physically demanding VR interface navigation and controls through any Unity-compatible VR headset and controllers or haptic devices.

Point clouds are a common way to visualise remote environments in 3D, but they often have distortions and occlusions, making it difficult to accurately represent objects' textures. This can lead to poor decision-making during teleoperation if objects are inaccurately represented in the VR reconstruction. A study using an end-effector-mounted RGBD camera with OctoMap mapping of the remote environment was conducted to explore the remote environment with fewer point cloud distortions and occlusions while using a relatively small bandwidth. Additionally, a tactile exploration study proposed a novel method for visually presenting information about objects' materials in the VR interface, to improve the operator's decision-making and address the challenges of point cloud visualisation.

Two studies have been conducted to understand the effect of virtual world dynamic scaling on teleoperation flow. The first study investigated the use of rate mode control with constant and variable mapping of the operator's joystick position to the speed (rate) of the robot's end-effector, depending on the virtual world scale. The results showed that variable mapping allowed participants to teleoperate the robot more effectively but at the cost of increased perceived workload. The second study compared how operators used a virtual world scale in supervised control, comparing the virtual world scale of participants at the beginning and end of a 3-day experiment. The results

showed that as operators got better at the task they as a group used a different virtual world scale, and participants' prior video gaming experience also affected the virtual world scale chosen by operators.

Similarly, the human-operator's visual attention study has investigated how their visual attention changes as they become better at teleoperating a robot using the framework. The results revealed the most important objects in the VR reconstructed remote environment as indicated by operators' visual attention patterns as well as their visual priorities shifts as they got better at teleoperating the robot. The study also demonstrated that operators' prior video gaming experience affects their ability to teleoperate the robot and their visual attention behaviours.

Acknowledgements

I would like to thank my supervisors Dr Ildar Farkhatdinov and Dr Maurizio Valle for their mentorship, support and encouragement throughout my PhD. Their expertise and patience have been invaluable in helping me to grow as a researcher and a person, and I am deeply grateful for the time and effort they have invested in me.

I would also like to thank my collaborators and co-authors. Their insights and contributions have greatly helped my research, and I am grateful for the opportunity to have collaborated with them.

I would like to acknowledge the Advanced Robotics Centre at Queen Mary University of London, and the University of Genoa's EMARO and COSMIC labs for providing me with the necessary funding and resources to carry out my research.

I would like to express my gratitude to my family, friends, and loved ones for their support and encouragement throughout my journey. I would not have been able to complete this program without their love and support.

Sincerely yours,

Bukeikhan Omarali

Statement of originality

I, Bukeikhan Omarali, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged and my contribution indicated. Previously published material is also acknowledged. I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material. I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis. I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university. The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Contents

List of Figures

List of Tables

Acronyms

AR Augmented Reality.

CNN Convolutional Neural Network.

GBN Gesture-based navigation.

GUI Graphical User Interface.

IMU Inertial Measurement Unit.

ITP Interoperable Teleoperation Protocol.

MCNN Multi-modal Convolutional Neural Network.

MR Mixed Reality.

PCA Principal Component Analysis.

RF Random Forest.

ROS Robot Operating System.

SLAM Simultaneous Localization And Mapping.

URDF Unified Robotics Description Format.

VR Virtual Reality.

XR Extended Reality.

1 Introduction

Contents

1.1 Thesis scope

Robot teleoperation is used for tasks that cannot be performed directly by a human or fully automated robots. There could be a number of reasons why a human cannot perform a task: the task-associated environment can be too dangerous for a human: nuclear power plants, deep underwater or in space; or a human might not be able to perform the task due to space and accuracy/precision limitations like in minimally invasive surgeries. Although artificial intelligence has been rapidly developed over the past decade, fully automated robots still are incapable of dealing with complex, unstructured environments and at best require human supervision - a "specialist in the loop" for high-level decisions and at worst a direct teleoperator "pilot" for moment-to-moment actions.

In the past few years teleoperating a robot with the help of Virtual Reality (VR) technologies has been steadily gaining popularity. Using VR as a proxy between the human-operator (from now on simply "operator") and the robot in the remote environment is by no means a novel idea - both robot teleoperation and virtual reality technologies have been around for more than half a century and attempts were made to merge them. But only recently VR and related technologies have become both sufficiently mature and affordable to be widely adopted in robot teleoperation - we have seen improvement in the graphical fidelity of VR headsets, motion tracking, simultaneous localisation and mapping, and depth cameras in VR sets commercially available for researchers.

In comparison to conventional robot teleoperation interfaces (based on 2D displays, keyboards joysticks or haptic devices), VR-based interfaces (headset and handheld wireless controllers) provide an operator with improved spatial perception, more intuitive control and remote environment exploration [1, 2]. VR headsets provide the operator

with a binocular vision - allowing them to view the remote environment in 3D rather than in 2D. VR headsets and handheld controllers also map their head and hand movements into the VR world, allowing them to physically move in the VR world, which improves their ability to explore and interact with the remote environment reconstructed in VR.

Virtual Reality (VR) is emerging as the primary method for robot teleoperation due to its potential to provide immersive and intuitive control interfaces. However, some challenges such as the reliability of remote environment 3D reconstruction and mapping using RGBD cameras, as well as human motion capture issues like tracking loss, still need to be addressed for VR-based teleoperation to become more robust. Nevertheless, even in its early stages, VR-based robot teleoperation has shown promising results and is outperforming traditional teleoperation methods. Further research and development in the field of VR-based robot teleoperation are warranted, as there are opportunities for novel findings and advancements in robot control interfaces and user experiences within the VR environment. Therefore, contributing to the development of VR-based robot teleoperation is a worthwhile endeavour.

Despite the increasing popularity of VR-based robot teleoperation, this field is still in its early stages of development, and there is a lack of clarity on how traditional control and visualisation techniques can be effectively transferred and adapted for VR-based human-robot interfaces. To address this gap, this thesis aims to investigate these aspects of VR-based robot teleoperation. Specifically, the research will focus on remote environment visualisation, the interaction between remote environment visualisation and robot control, and the dynamics of the operator's interaction with the VR interface. By examining these key areas, this thesis seeks to contribute to the advancement of VR-based robot teleoperation techniques. The research will explore how remote environments can be effectively visualised and mapped in VR using RGBD camera(s), how this visualisation and its' virtual-to-real scale impacts robot control, and how the operator's interaction with the VR interface can enhance the overall teleoperation experience. By shedding light on these critical aspects, this thesis aims to provide insights and recommendations for optimizing VR-based robot teleoperation, leading to more effective and intuitive control interfaces for robots in remote environments.

The following limitations are set in order to keep the scope of the research manageable. Studies are limited to VR-based robot teleoperation applications although other Extended Reality (XR) options - immersive frameworks that use a combination of real

and virtual elements, will be covered in the literature review, as some interaction techniques used there are applicable to VR. Similarly, the research is limited to robotic arms although interaction methods with mobile robots are included in the literature review. Time delay in robot teleoperation is also largely ignored. The research does not focus on any particular teleoperation application field, instead, it deals with general robot teleoperation in unstructured environments and assumes no a priori knowledge of the remote environment.

1.2 Research questions

In order to leverage the benefits of binocular vision that VR provides for robot teleoperation the visual information needs to be presented to the operator in binocular format, i.e. two images of the same scene captured by two cameras in interocular distance - the distance between human eyes. In the context of VR-based robot teleoperation, the most efficient method to achieve this is to capture the remote environment in 3D, reconstruct it in VR and render separate images of it for each eye. This is commonly achieved using RGBD cameras that can capture the remote environment in 3D and display it in VR as point clouds. Although a number of works have demonstrated that this method is sufficient to teleoperate a robot from VR [3, 4], a number of challenges remain: distortion of point clouds, point clouds' occlusion, RGBD cameras' limited field of view. These issues need to be resolved in order to maximise the remote environment presentation clarity to the operator in VR as it can affect their decision-making and the success of teleoperation.

Once the remote environment is properly reconstructed in VR, the next question becomes: "How can an operator manipulate the virtual world to the benefit of robot teleoperation?". As mentioned before, motion tracking can be used to allow the operator to physically navigate and interact with the virtual world. For example, the operator can physically walk around the VR reconstruction of the remote environment and use hand movements to control the robot. These are relatively "mundane" real world inspired modes of interactions; given that the virtual world can be freely manipulated there should be more interactions that are not subject to real world limitations that can benefit robot teleoperation and alter how common teleoperation techniques are used in VR.

Consider the scale of remote environment reconstruction in VR - the proverbial low-

hanging fruit of non-real interactions. Unlike the real world, in a virtual world the remote environment can be scaled up or down, i.e. similar to Alice in Wonderland, the operators can make themselves bigger or smaller relative to the virtual world. These interactions can have different implications on teleoperation flow and human-robot interactions. One deceptively obvious hypothesis is that given the ability to scale the virtual world, the operator would manipulate the scale depending on the robotic task at hand, for example: the operator can zoom-in to inspect an object or perform a precision movement. However, it is yet unclear whether it is an optimal operator behaviour. Or consider the following. One common technique in robot teleoperation is scaled mapping of input device movement onto the robot in direct position-position control. This is used when the workspace of the input device is smaller than that of the robot, for example: a 1cm movement of the small haptic input device can become a 10cm movement of the robot. Changing the virtual world scale can have a similar effect - the operator scales down the virtual world (i.e. zooms-out) and a small movement of the input device in the real world becomes a large movement of the robot in the virtual world. On the other hand, in rate control (position-velocity mapping) the displacement of the input device is mapped to the robot's velocity, like the gas pedal of a car. Similar to scaled movement mapping this is a control method designed to deal with size mismatch between the input device's and the robot's workspace. However, the interaction between the virtual world scale and rate control has not yet been studied.

Lastly, the thesis aims to improve our understanding of how operators teleoperate robots from VR in terms of actions and visual priorities. What is the difference between a "good"/expert and a "bad"/novice teleoperator? Would they use different virtual world scales, or spend more/less time looking at the manipulation objects? Understanding how operators use VR to teleoperate robots can be used to improve how we design VR-based robot teleoperation interfaces.

Hence my research questions can be summarised as follows:

1. What are the most efficient methods for visual representation and exploration of the remote environment for VR-based robot teleoperation?

2. How does the VR visualisation scale of the remote environment affect the human-operator's ability to control the robot?

3. How do naive users learn to teleoperate a robot in VR and how do their actions and visual attention priorities change during learning?

1.3 Research contributions and publications

The research contributions of this thesis pertain to the previously stated research questions and have been published in peer-reviewed conferences and journals. Specifically, novel methods for remote environment exploration, navigation, and visualisation for VR-based robot teleoperation have been proposed. Additionally, the research has led to a greater understanding of the effects of virtual world scale manipulation on the operator's ability to control a robot, as well as the learning process and visual priorities of operators during VR-based robot teleoperation. A comprehensive VR-based robot teleoperation framework, incorporating both existing and novel features, has been developed and has been utilised in experiments presented in this thesis and others conducted by colleagues in related fields. This framework serves as a valuable starting point for future research in VR-based robot teleoperation.

In summary the contributions of this thesis are the following:

- Developed a novel VR-based robot teleoperation framework.
- Investigated methods for remote environment visual reconstruction and exploration in VR.
- Developed a novel method for classifying objects' materials and presenting this information visually in the VR interface.
- Investigated effects of virtual world scale on operator's ability to teleoperate the robot in different control modes.
- Investigated operator's visual priorities during VR-based robot teleoperation and their dependency on operator's experiences.

The research presented in this thesis resulted in following papers:

- 2022, **B. Omarali**, M. Valle, K. Althoefer, I.Farkhatdinov *"Visual Attention in Virtual Reality based Robot Teleoperation"*, planned submission to Science Robotics
- 2022, **B. Omarali**, S. Javaid, M. Valle, I.Farkhatdinov *"Workspace Scaling in Virtual Reality based Robot Teleoperation"*, under review at 2023 ACM Augmented Humans (AHs) International Conference

- [5] 2022, **B. Omarali**, F.Palermo, M. Valle, K. Althoefer, I.Farkhatdinov *"Tactile Classification of Object Materials for Virtual Reality based Robot Teleoperation"*, 2022 IEEE International Conference on Robotics and Automation (ICRA 2022)
- [6] 2021, **B. Omarali**, M. Valle, K. Althoefer, I.Farkhatdinov *"Workspace Scaling and Rate Mode Control for Virtual Reality based Robot Teleoperation"*, 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
- [7] 2021, I. Vitanov, I. Farkhatdinov, B. Denoun, F. Palermo, A. Otaran, J. Brown, **B. Omarali**, T. Abrar, M. Hansard, C. Oh, S. Poslad, C. Liu, H. Godaba, K. Zhang, L. Jamone, K. Althoefer *"A Suite of Robotic Solutions for Nuclear Waste Decommissioning"*, Robotics
- [8] 2020, **B. Omarali**, B. Denoun, M. Valle, K. Althoefer, I.Farkhatdinov *"Virtual reality based telerobotics framework with depth cameras"*, 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)
- [9] 2019, **B. Omarali**, F. Palermo, M. Valle, S. Poslad, K. Althoefer, I. Farkhatdinov *"Position and velocity control for telemanipulation with interoperability protocol"*, Annual Conference Towards Autonomous Robotic Systems (TAROS)

1.4 Thesis structure

The thesis has a following structure:

- Chapter 1 presents the motivation and goals for this thesis, and it also presents a literature review on VR-based robot teleoperation-based teleoperation research, with a focus on remote environment visualisation and robot control.
- Chapter 2 presents the VR-based robot teleoperation framework developed during the research and used in subsequent studies. The framework defines the leader-follower communications, remote environment visualisation methods from RGBD cameras and the robot's state, and a set of control techniques and gestures used by an operator to perform telemanipulation.
- Chapter 3 presents two studies on remote environment visualisation in VR-based robot teleoperation. The first study inspects the effects of RGBD camera place-

ment on the operator's ability to visually explore and understand the remote environment. The second study presents a method for tactile exploration-based classification of objects' materials that are visually communicated to the operator.

- Chapter 4 presents a study on the effects of the remote environment's virtual reconstruction scale on the operator's ability to control the robot in rate mode.
- Chapter 5 addresses human-centred behavioural aspects of the proposed VR-based teleoperation framework. Gaze tracking is used in a participant study to analyze changes in visual attention as novice participants learn to teleoperate a robot from VR. Furthermore, participants' usage of the virtual world scale is investigated. The results provide valuable insights into how VR-based teleoperation interfaces should be designed.
- Chapter 6 Summarises main achievements, discusses the limitations of the studies, and outlines potential future work directions.

1.5 Related works

This section presents a literature review on state-of-the-art VR-based robot teleoperation and related fields. First, the brief history (1940s - early 2000s) of robot teleoperation and VR technologies is presented. Next, existing immersive teleoperation interfaces - stereoscopic telepresence, Augmented Reality (AR), VR, are reviewed, and reasons to focus on VR-based robot teleoperation are presented. Then, existing VR-based robot teleoperation interfaces and methods are discussed in detail with a focus on the integration of VR technologies and robot teleoperation, remote environment visualisation and robot control. Finally works that contextualise the human gaze for VR-based robot teleoperation are discussed.

1.5.1 Brief history of robot teleoperation and VR

It could be argued that the concept of teleoperation existed as far back as fire tongs, animal prods and other simple arm extensions and actuated prosthetic limb fitters. However, in the context of this research, it is important to note the emergence of modern teleoperation in the 1940s, when Goertz introduced the first "master-slave" device at

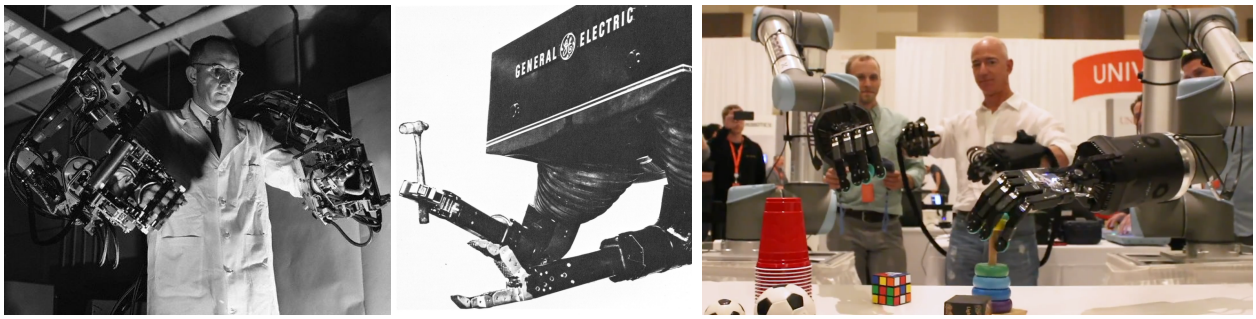


Figure 1.1: An illustration of nearly 60 years of robot teleoperation progress. Left - the 1967 Handyman (General Electric) bi-manual robot teleoperation setup; right: the 2021 Shadow Robotics bi-manual robot teleoperation setup.

Argonne National Laboratory near Chicago [10]. This was a purely mechanical pantograph mechanism that directly mapped the Cartesian position and rotation of the input device to those of a follower device in real time. The operator could manipulate radioactive materials in a "hot cell" safely from outside the cell. The mechanical coupling of the leader and the follower devices allowed the operator to feel the force exerted by the follower device in the remote environment. Nowadays in technical literature and further in this paper such control method is referred to as real-time direct bilateral control - a robot is teleoperated in real-time with direct mapping of the leader's pose to the follower's pose whilst providing force feedback. Goertz soon replaced the direct mechanical tape and cable linkages with electrical servomechanisms and introduced a closed-circuit television so that the operator could be an arbitrary distance away.

The next noteworthy development was the 1967 Handyman system presented by Mosher [11] for the General Electric Co, see Fig. 1.1 left. The Handyman was a stepping stone towards the Hardiman exosuit, and its leader side was designed as a wearable arm harness that controlled a bi-manual (two-arm) robot. Similar to Goertz's system the Handyman was fully electrified and its' two master devices compensated for their own gravity and provided force feedback. Although the Handyman was a breakthrough for intuitive human-centric robot teleoperation it was too expensive and bulky to be used in the field, meanwhile, the development of Hardiman did not go past the prototype stage citing: "violent and uncontrollable motion by the machine".

In 1966 the US NAVY successfully teleoperated a Cable-controlled Undersea Recovery Vehicle (CURV) to find and retrieve a nuclear bomb from the deep ocean bottom,

accidentally dropped from an airplane off Poiomares, Spain [12]. While the original design was fairly simplistic, its successor, CURV-III was equipped with continuous transmission frequency modulated (CTFM) sonar sensors, TV cameras, digital cameras and two manipulators; it was further used for similar search and rescue operations. Additionally, offshore mineral extraction and cable-laying firms soon became interested in this technology to replace human divers, especially as oil and gas drilling operations got deeper.

In the early 1960s the race to the moon began. The time delay between the moon and earth proved to be problematic for previous real-time closed-loop control as it resulted in instability demonstrated by the "Surveyor" lunar landers and the Lunokhod rover. In 1965 Ferrel et al. [13], demonstrated that an open-loop "move-and-wait" approach - the operator makes a move then waits for a confirmation before making the next move - is more efficient than direct control in the presence of time delay. A follow-up study [14] discussed further independence and automation of follower robots - how the percentage of decisions that the remote robot makes affects the effectiveness of teleoperation. Control approaches where a considerable percentage of low-level decisions (for example motion planning) are made by the robot are referred to as supervised control. Robot teleoperation developed further in incremental steps finding applications in hazardous material handling, robotic surgery, etc. Now let us examine the history of VR.

Similar to robot teleoperation it can be argued that early VR existed as far back as panoramic paintings or simple stereographic photo viewers, however, the concept of goggles that let the wearer experience a fictional world through holographics, smell, taste and touch was first introduced by science fiction writer Weinbaum in 1935 "Pygmalion's Spectacles" [15]. In the 1950s the concept was brought to life (to a reasonable extent) by Heilig's Sensorama [16] - an arcade-style theatre cabinet that would stimulate all the senses, not just sight and sound. It featured stereo speakers, a stereoscopic 3D display, fans, smell generators and a vibrating chair.

In 1961, two Philco Corporation engineers (Comeau & Bryan) developed the first precursor to the head-mounted displays as we know them today – the Headsight [17]. It incorporated a video screen for each eye and a magnetic motion tracking system, which was linked to a closed-circuit camera. The Headsight was not developed for immersive remote viewing of dangerous situations by the military. Head movements would move a remote camera, allowing the user to naturally look around the environment. Headsight was the first step in the evolution of VR head-mounted displays but it lacked the

integration of computer-based image generation.

In 1968 Ivan Sutherland and his student Bob Sproull created the first VR head-mounted display "Sword of Damocles" [18] that was connected not to a camera but to a computer that generated primitive wireframe rooms and objects. Similar to its namesake it was an imposing contraption suspended from a ceiling too heavy for the user to wear comfortably. 1990s NASA, with the help of Crystal River Engineering, created Project VIEW [19] - a VR-powered simulator used to train astronauts. VIEW looks recognizable as a modern example of VR and features gloves for fine simulation of touch interaction.

In 2004 Kron et al. [20] presented a proof of concept setup that successfully combined static stereoscopic vision and direct robot control. In 2005 Wai-keung et al. [1] demonstrated that 3D stereoscopic visualisation of the remote environment outperforms 2D monoscopic (2D display) visualisation in robot teleoperation. In 2000 the Da Vinci [21] - a robotic surgical system for minimally invasive surgery with a stereoscopic operator's interface was developed.

1.5.2 Immersive robot teleoperation interfaces

To avoid possible confusion, the use of the term "virtual reality" in robotics literature and the thesis needs to be addressed. There are two things of note. First - some papers use the term "virtual reality" to refer to any simulated (not "real", hence "virtual") interaction. For example, model-mediated teleoperation uses models of the robot and the environment to predict and render force interactions between the robot and the environment as the means of alleviating the latency issues - in this context, authors use the term "virtual reality" to refer to the simulation. Second - a misuse of XR, AR, MR, VR taxonomy by authors. Extended Reality (XR) is an umbrella term for AR, MR, and VR. Augmented Reality (AR) provides the view of the real world with an overlay of digital elements, for example, Microsoft Hololens applications or AR applications on smartphones. Virtual Reality (VR) provides the view of a fully immersive digital environment, for example, VR games. Mixed Reality (MR) originally was used as an umbrella term for AR and VR, but lately is used to refer to the middle-ground between AR and VR, where virtual and real elements can interact with each other. It is not uncommon for some authors to refer to AR applications as VR, or VR as MR.

For the purposes of the thesis existing XR-based robot teleoperation research is split depending on the user interface immersiveness as illustrated in Fig. 1.2. On the left side

of the spectrum (omitted in the figure for readability) one would find the likes of Goertz's original master-slave device and conventional 2D display, keyboard and mouse robot teleoperation setups; on the right side - a fully simulated VR-based teleoperation environment, for example, [22]. Meanwhile, in between, there are spectroscopic telepresence, collocated AR, and VR-based robot teleoperation setups. It should be noted that a considerable number for works reviewed from now on are contemporary to the research presented in this thesis - some methods discussed below were developed at the same time or after the author's work.

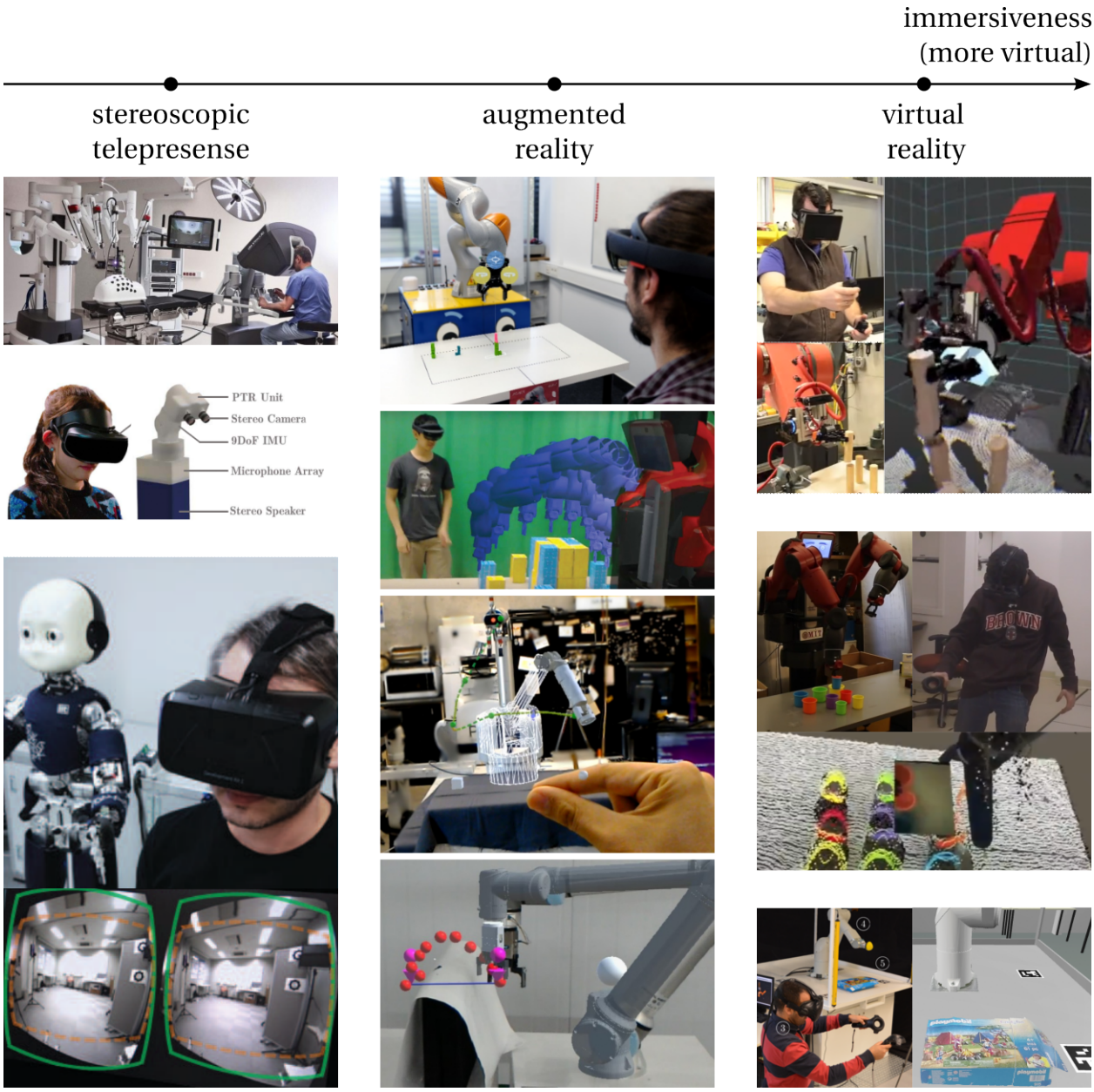


Figure 1.2: Examples of existing immersive teleoperation setups from least to most virtual, columns left to right: stereoscopic telepresence [23, 24, 25], augmented reality based robot teleoperation [26, 27, 28, 29], virtual reality based robot teleoperation [30, 4, 3].

1.5.2.1 Stereoscopic telepresence

Stereoscopic telepresence devices are arguably less immersive and least virtual compared to the other two groups - as they only use virtual reality headsets to render stereo video. As a rule of thumb, the stereo video is captured by a stereo camera - a camera that captures two (or more) views that are an intra-ocular distance away from each other, such that they can be viewed one-view-per-eye by the operator. The binocular vision allows the operator to perceive the depth of the remote environment, i.e. see the remote environment in 3D. The most known example of such a setup is the Da Vinci surgical robot, see Ballantyne et al. [23]. In simplest terms, the Da Vinci system can be thought of as two robotic arms controlled bilaterally by two haptic devices, with a controllable stereo camera and a stereo display. One interesting aspect of the Da Vinci is the camera control. Camera control is particularly important in remote telepresence - the human brain expects the view to adjust according to head movements, resulting in motion sickness otherwise. Da Vinci constrains the operator's head movements - from the operator's perspective, the operator looks through a static window directly at the remote environment. In order to control the camera the operator needs to engage the camera control by pressing down a corresponding pedal and moving the haptic devices. From the operator's perspective, it is not the camera that is moving but the remote environment in the window. Although this setup reduces motion sickness, one major disadvantage is that robotic arm control and camera control do not work simultaneously. This problem was addressed by Dardona et al. [31] that proposed a method for controlling the camera using a VR headset's head tracking.

The concept of head tracking-based control of the remote stereo camera is fairly common in telepresence robots. Kohlbecher et al. [32] proposed a "Wizard-of-Oz" setup where the robot's head-mounted camera mimics the operator's head and eye rotation with a camera feed shown on a single 2D screen. Tran et al. [33] proposed a novel method for generating stereoscopic presence by splitting a single monoscopic camera view of a robot per eye. Theofilis et al. [25] extended the concept further for VR headsets by rendering iCub robot's eye cameras on the headset's separate eyes. Martins et al. [34], Doisy et al. [35], and Mackey et al. [36] have used VR headset's tracking to control actuated stereo cameras in the remote environment. Aykut et al. [24] utilised deep learning techniques for motion prediction in teleoperation to reduce motion sickness originating from a mismatch of human and robot head poses. Parasuraman et al. [37] displayed the video from a 360-camera projected on the inside of a sphere presented

to the teleoperator in VR for virtual tours. A number of other works have also utilised VR as virtual "control rooms" with a stereoscopic window into a remote environment [38, 39, 40, 41].

Stereoscopic telepresence systems rarely use virtual elements, with the exception of the Lipton's et al. [42] "homunculus" framework that uses virtual interaction proxies in between the operator's input device and the robot. The operator is placed in a control room "inside" the robot's head with a stereoscopic window to the remote environment and interaction proxies that the operator can manipulate in order to move the robot.

1.5.2.2 AR-based robot teleoperation

AR-based robot teleoperation displays virtual elements on top of the real world, for example, desired end-effector poses for motion planning can be set and visualised as virtual elements on top of the real world robot. AR-based telerobotics setups are often collocated - the operator needs to be able to directly see the robot and the robot's environment with their own eyes. An exception is work by Yew et al. [43] that blends virtual reconstruction of a priori known elements from the remote environments and the robot with physical doubles for maintenance applications.

Most AR-based setups use a Microsoft Hololens headset as it provides operator's head and hands tracking, gesture and speech recognition as well as Simultaneous Localization And Mapping (SLAM) of the local environment. These two features lend themselves well to supervised teleoperation - the operator can use gestures and speech to place a number of markers with respect to the environment's geometry for the robot's motion planning, preview the motion plan and then execute. To the author's best knowledge, AR is not used with real-time control due to inaccurate moment-to-moment hand tracking. Quintero [28] et al. and Guhl et al. [29] demonstrated how a collocated AR-based interface can be used to plan robotic motions along a curved surface. Krupke et al. [44] demonstrated the effectiveness of AR interface in pick-and-place tasks, extended by Hernandez et al. [45] to chain multiple high-level tasks (multiple pick-and-places for example) and Rosen et al. [27] to generate action-oriented semantic maps, for example, "light switch" can be "flipped".

There have been several studies that have explored the use of visualisation techniques in AR for teleoperation. Rosen et al. [46] proposed the use of AR to preview the motion of a robot in collocated, collaborative setups, which can improve the efficiency and safety of the task. Similarly, Walker et al. [47] demonstrated that redirecting the

user's input to an AR virtual surrogate robot, which foreshadows the physical robot's actions, can lead to improved performance compared to direct control of a physical robot using a joystick. Ostanin et al. [48] introduced the concept of virtual doubles, which are segmented parts of real environments that can be scaled to enable the more precise positioning of the robot's end-effector. Furthermore, Arevalo et al. [26] suggested the use of visual cues in AR for teleoperation in collocated robot setups, which can improve performance by providing a priori knowledge of the remote environment.

1.5.2.3 VR-based robot teleoperation

VR-based robot teleoperation places the operator into an entirely digital environment. As this thesis focuses on VR-based robot teleoperation only one example of VR-based robot teleoperation is mentioned here (with a lot more to follow below) - the Guerin's et al. [30] Adjutant framework. Although VR was not the focal point of the framework a few key features of VR-based robot teleoperation were demonstrated there: visualisation of the remote environment using point cloud collected from an RGBD camera in the remote environment, a robot represented in VR as a 3D mesh animated by the real-time feedback of the robot, the operator uses 6-DOF wireless controllers to set the robot's desired end-effector pose, virtual fixtures - user-generated virtual 3D geometries that constrain movements of the robot.

Of the three groups presented VR-based robot teleoperation is the most flexible. Unlike stereoscopic telepresence VR-based robot teleoperation decouples the operator's view and the robot's camera view, i.e. without moving the camera, the operator can look at the remote environment from different angles by navigating the virtual world. Unlike AR-based robot teleoperation, VR-based is suitable for bilateral direct control of remote robots. Furthermore, VR allows the operator to manipulate the virtual world reconstruction of the remote environment and implement a wide range of virtual elements that can help teleoperate the robot.

It should also be noted however that VR-based robot teleoperation also has the most technical challenges to be resolved. To name but a few: point cloud based visualisation of the remote environment can often appear distorted or occluded by the robot; interactions between remote environment visualisation and robot control need further studies; it is yet unclear how to best navigate the virtual world, etc.

1.5.2.4 Summary of immersive robot teleoperation interfaces

Table 1.1 summarises immersion teleoperation interfaces. In stereoscopic telepresence, the operator's view is limited to the stereo camera view. In order to explore the remote environment the camera needs to be moved either manually or using head tracking although the latter is known to cause motion sickness. Stereoscopic telepresence has been shown to be effective in minimally invasive surgeries and virtual tours. AR-based robot teleoperation lends itself well to collocated supervised control of robots but not so much to real-time control of remote robots. AR-based robot teleoperation can be used for collaborative robotics tasks as well as intuitive robot programming. VR-based robot teleoperation is the most flexible of immersive teleoperation options but also the one with the most technical challenges to be resolved. VR-based robot teleoperation is also most suited for remote robot control, therefore, from now on the literature review will focus on VR-based robot teleoperation.

Table 1.1: Immersive teleoperation interfaces: applications and limitations

mode	limitations	applications
Stereoscopy	• operator's field of view is limited to the stereo camera view; • can cause motion sickness if stereo-camera is not as dexterous as operator's head movement tracking.	• minimally invasive robotic surgery; • virtual tours.
AR	• limited to collocated human-robot interaction scenarios as virtual elements need to be layered over the real world;; • limited to supervised robot control due to current hardware limitations.	• collocated human-robot interaction.
VR	• faulty remote environment reconstruction in VR using RGBD sensing; • unclear how VR-interface can affect robot control; • VR reconstruction manipulations are still unexplored.	• remote robot control.

1.5.3 Integration of VR technologies with robot teleoperation and ROS

Before delving further into the review of visualisation and control in VR-based robot teleoperation, the integration of VR technologies and ROS needs to be discussed as the majority of works reviewed use Robot Operating System (ROS). To this day there is very little integration between commercially available VR sets and ROS. ROS is an open-source software platform that is widely used for building and operating robots. It provides a set of tools, libraries, and conventions for building and deploying robot software. In early releases of modern VR sets, manufacturers provided little support for Linux-based systems preferring Windows - the primary operating system for VR gaming. Hence, aside from a few exceptions [49, 50] most VR-based robotics researchers were forced to adopt Windows-based video game engines as middleware for VR integration.

A video game engine is 3D video game development software, responsible for 3D rendering of the game used to speed up and streamline the development. A number of video game engines are freely available for non-commercial research purposes: Ktena et al. used [51] the Unreal Engine and Omarali et al. [52] used the Blender game engine. The Unity game engine is the most popular amongst robotics researchers as it is relatively beginner-friendly, supports development for AR and VR applications and allows researchers to integrate outside libraries like custom Inverse Kinematics solvers [53, 54] or drivers for haptic devices, etc.

Most Unity-based teleoperation interfaces use ROS-styled communications facilitated by ROS-bridge introduced by Crick et al. [55]. ROS-bridge is a set of software libraries that provide an interface between ROS and non-ROS programs, allowing non-ROS programs to communicate with ROS programs and access their functionality and data. ROS-bridge is implemented as a set of ROS nodes that communicate with non-ROS programs using various network protocols, such as WebSockets or JSON-RPC.

Whitney et al. [4] introduced a ROS-Reality framework that utilised ROS-bridge to facilitate communications between the Unity-based leader and ROS follower setups. This framework demonstrated how VR features mentioned in Adjutant [30] (the Adjutant was proposed prior to ROS-bridge development) can be implemented in Unity using ROS-type communications. The ROS-reality framework used a custom JSON-based ROS publisher/subscriber on the Unity-leader. The framework was used to perform various manipulation tasks (for example cup stacking task) and outperformed a con-

ventional 2D monoscopic display, keyboard and mouse setups in [2].

Currently, the most efficient implementation of ROS-bridge communication between Unity and ROS is the ROS# library. It is a set of software libraries and tools for building robot applications using the C# programming language and the .NET framework and is available as a Unity package providing base classes for publisher, subscriber, action and service client, etc.

1.5.4 Remote environment visualisation in VR

This section aims to discuss the current state-of-the-art in the field of remote environment capture, and visualisation in VR, as well as the control of cameras and the operator's view in teleoperation. First, the body of literature in this field is divided into two main categories: works that deal with unstructured and structured remote environments. These corresponding works often have different applications and sets of assumptions and constraints. Next methods for capturing and reconstructing the remote environment in VR are discussed, followed by visualisation methods of objects' properties in VR and methods for navigating the virtual world.

1.5.4.1 Unstructured and structured remote environments

An unstructured remote environment is one that is not known a priori, i.e. before teleoperation begins the operator and the robot have no prior knowledge of what objects may be in the remote environment, where are they located, etc. DARPA robotics challenge that simulated a disaster clean-up is a good example of an unstructured remote environment [56, 57]. Unstructured environments often are captured and visualised in VR as point clouds, discussed in more detail further below.

Logically a structured remote environment is at least partially known a priori. Hence they are often found in applications with clearly defined goals or simulations. For example, Mae et al. [58] presented VR-based robot teleoperation of a multi-legged robot over a known terrain. Vagvolgyi et al. [59] overlayed images and textures from the real world on 3D models in the virtual world to simulate the control of space robots. Horikawa et al. [60] mapped a real world known environment completely to VR, effectively becoming an AR application such that robot's intentions can be previewed similar to [46]. Structured environments are usually represented in VR using 3D meshes.

Structuring the remote environment is beneficial in terms of communication and

computation load as well as visualisation clarity. Communicating point clouds require large bandwidth, so replacing some of the objects with pre-defined 3D meshes can reduce the bandwidth consumption dramatically. For example, Kohn et al. [3] removed point clouds of remote environment background and robot, such that only manipulation objects were represented as a point cloud, reducing the communication average from 760MB/s to 4MB/s. Similarly, Feichter et al. [61] simplified the point cloud by detecting planes like floors or walls. Zhou et al. [62] demonstrated how the point cloud of known objects can be segmented and replaced with a corresponding meshes. It needs however be stressed that methods described above require some a priori knowledge of remote environment which could be impractical in scenarios like disaster relief. Reducing the size of point clouds is also beneficial for visual rendering as large point clouds are computationally expensive to render in VR [63]. Another option is to downsample the remote environment point cloud by voxelising it and representing the remote environment as an OctoMap [64].

1.5.4.2 Capturing the remote environment

The most popular (and simplest to implement) method for capturing the remote environment as a point cloud is using a RGBD camera. Alternative methods for point cloud generation are from multiple-RGB images as demonstrated by Wu et al. [65] or omni-camera - Thomason et al. [66] or tactile exploration - Luo et al. [67] and Izatt et al. [68].

A RGBD camera is similar to RGB video camera, except it also has an infrared emitter and receiver that allows the camera to estimate the depth of the scene. Hence combining both the video stream (RGB) and the depth (D) one can generate (a process often called in literature "register") the point cloud of the remote environment. At the current state of technologies point cloud registration has issues, as discussed by He et al. [69]. Here the ones relevant to VR-based robot teleoperation are mentioned. Depth measurements do not work well on smooth metallic surfaces - point cloud registration can generate a lot of noise (incorrectly registered points in point clouds) and filtering algorithms are computationally expensive [70]. Point clouds also can appear distorted to the point that the original object is unrecognizable. Point clouds provide little information about the objects' texture and materials.

The majority of VR-based robot teleoperation setups employ a single RGBD camera, resulting in a self-occluded point cloud of the remote environment - the camera can only see one side of the remote environment. Multiple cameras that look at the remote

environment from different positions and angles can generate a more complete reconstruction of the remote environment as demonstrated by Kohn et al. [3] and Wei et al. [71]. Alternatively, multiple cameras can be placed at some positions, looking in different directions producing a near 360-degree reconstruction of the remote environment as demonstrated by Okura et al. [72, 73], albeit self-occluded. However using multiple camera setups require larger communication bandwidth and robust methods for calibrating cameras' extrinsic parameters - i.e. finding their relative poses in space [74, 75]. One can consider a dynamic camera. To the best of the author's knowledge, there have not been any studies on automated/dynamic cameras for VR-based teleoperation, however, a few methods from conventional teleoperation can be applicable. Nicolis et al. [76] and Rakita et al. [77] proposed methods for robot end-effector mounted RGBD camera automated control for two robotic arm setups. Similarly, Draelos et al. [78] and Su et al. [79] proposed methods for dynamic/automated RGBD camera placements for robotic minimally invasive surgeries.

1.5.4.3 Detecting objects' material properties

As discussed above point cloud based reconstruction of the remote environment is far from perfect - for example, point clouds have problems visualising objects' materials. Hence one can consider alternative methods that could increase what information the operator has. To the best of the author's knowledge, there have been no works that investigate object material visualisation for VR-based robot teleoperation. However, findings in related fields can be applicable.

The classification of material properties of objects using machine learning applied to images is a widely explored topic in the literature, as reported in various studies such as [80, 81]. One method commonly employed is the use of a Convolutional Neural Network (CNN) architecture to classify materials from patches extracted from photographs of objects, and subsequently localise and segment the materials in the original images, as demonstrated in [82]. However, in scenarios where lighting is limited and computer vision methods may prove ineffective, such as in teleoperation in extreme and hazardous environments [83], tactile exploration can serve as an alternative method for material recognition [84, 85]. This approach utilises a combination of proximity, tactile, and force sensing to provide important information about the explored material, such as texture, stiffness, and shape [86]. Furthermore, the compliance properties of various objects have been investigated using supervised classifiers with a hybrid force and

proximity finger-shaped sensor [87]. In addition, fibre optic-based sensors have been designed and used to recognise and classify fractures on surfaces using a random forest classifier [88, 89, 90].

Deep learning models have been shown to achieve high performance in various tasks, such as speech recognition [91] and image recognition [92], by leveraging high-dimensional inputs. In contrast, traditional machine learning algorithms rely on engineered features extracted from the data. In a study by Baishya et al.[93], it was found that a deep learning model for tactile material classification outperformed traditional machine learning classifiers such as Gaussian, K-nearest neighbours, and support vector machines when applied to data obtained from a tactile skin sensor attached to an iCub humanoid robot. This is due to the ability of deep learning models to learn from the raw high-dimensional data, whereas traditional machine learning algorithms require engineered features to perform well. Additionally, Gao et al.[94] demonstrated that using a multi-modal approach combining visual and physical interaction signals with CNNs led to more accurate results than using vision alone, and outperformed traditional feature-based methods. Furthermore, Alameh et al. [95] proposed an algorithm that uses human interactions on an electronic skin to recognise objects. The 3D tactile data generated by the skin was converted into 2D images and used as input for a CNN, which surpassed the performance of traditional tactile data classification algorithms.

1.5.4.4 Navigating in VR

Finally, it is important to consider how an operator can navigate and explore the remote environment reconstruction in teleoperation. Various methods exist for navigating the VR space, but only a few of them are suitable for use in robot teleoperation. For example, joystick control has been known to stimulate motion sickness [96], transportation methods can leave the operator feeling disoriented [97], and physical walking requires large "safe" spaces and can result in the operator getting entangled in the cables of the VR headset. Physical "walking in place" [98] is sensitive to the accurate mapping between leg movements and camera movements and Omnidirectional platforms are bulky and expensive. Therefore, it is essential to consider the limitations and trade-offs of different methods when designing navigation and exploration interfaces for teleoperation systems.

1.5.4.5 Summary of remote environment visualisation in VR

Unstructured remote environments are those that are not known a priori, for example, consider disaster relief scenarios. They are often visualised in VR as point clouds. Structuring these environments can improve communication, computation load, and visualisation clarity. Communicating point clouds requires large bandwidth, so replacing some known objects with pre-defined 3D meshes can reduce bandwidth consumption. However, this approach requires some a priori knowledge of the remote environment, which can be impractical in real world scenarios.

Point clouds of remote environments are commonly captured using RGBD cameras, but can also be generated from multiple RGB images, omnicamera, or tactile exploration. Single RGBD cameras are typically used in VR-based robot teleoperation setups, resulting in a self-occluded point cloud. Multiple cameras can generate a more complete reconstruction but require larger communication bandwidth and robust calibration methods.

Point cloud-based reconstructions of remote environments lack material information, so alternative methods should be explored. Machine learning methods have been used to classify material properties of objects from images, and tactile exploration can be used in low-light situations. Deep learning models have shown high performance in tactile material classification, outperforming traditional classifiers.

Different methods for navigating VR space exist, but only a few are suitable for robot teleoperation due to their limitations. For example, joystick control causes motion sickness, transportation methods disorient operators, physical walking requires large spaces and can result in tangled cables, and omnidirectional platforms are bulky and expensive. When designing navigation and exploration interfaces for teleoperation systems, it is essential to consider the limitations and trade-offs of different methods.

1.5.5 Robot control in VR-based robot teleoperation

Control methods in VR-based teleoperation can be broadly split into three categories: direct, shared and supervised control based on the level of human-operator's involvement. This section covers the advances in these methods as they pertain to VR-based robot teleoperation.

1.5.5.1 Direct control

Direct control refers to the process of directly and continuously specifying the actions and movements of a remotely-operated robot, for example continuously controlling the robot's end-effector pose or joint angles. Direct control can be split into unilateral (no force feedback) and bilateral (with force feedback).

In unilateral control, the input device's movements affect the robot's movements, but not vice-versa. In recent years, particularly in the context of VR-based robot teleoperation, human motion-capture-based control (also referred to as mimicry control) became popular - human movements are mapped onto the robot, for example, the robotic arm can mimic human arm movements. This kind of control has been found to be intuitive for operators compared to haptic devices or joysticks as demonstrated by Rakita et al. [99]. The human motion capture can be achieved either through inertial sensing - Rubagotti et al. [54], visual tracking - Kofman et al. [100], RGBD sensing - Reddivari et al. and Peppoloni et al. [101, 102].

VR handheld controllers are arguably the most popular option - controllers use a combination of inertial sensing and visual tracking and share integration tools with VR headsets; for the sake of brevity consider these notable examples [56, 4, 30]. Nearly all VR-based robot teleoperation works that use handheld controllers do semi-continuous movement mapping - the operator can pause the mapping, reposition the input device freely to better utilise its' workspace and re-engage the mapping. An extension of semi-continuous mapping is the "interaction proxy" introduced by Sun et al. [103] - instead of mapping the operator's movements onto the robot, movements of an interaction proxy, controllable by the operator are mapped to the robot. Interaction proxy's movements can be constrained (for example along a single axis) allowing the operator more precise control.

Since most robotic arms are not anthropomorphic human motion-capture-based control methods usually map the human's hand pose movement onto the robot's end-effector. During teleoperation, the operator can avoid hitting obstacles with the robot's end-effector but not with the rest of the robot's body. Hence it is important to detect and classify which objects in the remote environment are manipulatable objects that the robot can interact with and which are obstacles that the robot needs to avoid. If the remote environment is captured as a point cloud one also needs to be able to distinguish between the part of the point cloud that represents the robot and parts that represent other objects in the remote environment; for example, Su et al. [104] proposed a method

for segmenting the robotic tool from pointcloud. Once the obstacle is determined the robot needs to avoid it. Castro et al. [105] proposed a method for real-time collision detection that creates a 3D representation of detected external objects with depth cameras and stops the robot from possible collisions. Leeper et al. [106] and Rubagotti et al. [54] used single-step and multi-step optimisation-based model predictive controllers to teleoperate a robotic arm with static obstacle avoidance. Yang et al. [107] used a combination of pseudo-inverse Jacobian and deep learning to teleoperate a robotic arm with dynamic obstacle avoidance. Rakita et al. [108] proposed an optimisation-based IK-solver with deep learning for dynamic obstacle avoidance. Recently pure machine-learning-based IK solvers are becoming popular [109, 110], although they do not possess collision avoidance features yet.

It needs to be mentioned that remote environment segmentation is a broad and complex topic on its own, for example, consider the following works on in-door environment segmentation [111, 112, 113, 114, 115, 116, 117, 118]. Such segmentation is often performed on previously known objects using deep learning. Hence the applicability for unstructured environments is limited with a few exceptions, like the work of Danielczuk [119] for the segmentation of unknown objects in clutter.

The benefits of VR in bilateral control are more limited as the operator is more physically constrained in the real world, for example, the operator needs to always stand/sit in front of the haptic device, and cannot explore the virtual world by walking around [120]. To the author's best knowledge, VR has not had a significant effect on bilateral control methods. However, given that most VR-based teleoperation setups use some form of RGBD sensing, model-mediated bilateral control fits well with VR. In model-mediated bilateral control, the force feedback of interaction between the robot and environment is generated not from actual interaction but from a corresponding virtual model. These methods use parts of a point cloud near the robot's end-effector to estimate possible surfaces and generate force feedback accordingly, consider the following body works [121, 122, 123, 124, 125, 126, 127, 128]. Alternatively, the remote environment geometries can be analysed to generate virtual fixtures that are used to render force feedback, see [129, 130, 131].

1.5.5.2 Shared control

Shared control refers to a mode of operation in which the actions and movements of the robot are controlled both by the user and by the robot's onboard control system. To the

author's knowledge shared control is fairly uncommon in VR-based robot teleoperation.

In conventional teleoperation shared control often takes the form of assistive control, where the robot may either adjust the operator's input [132, 133, 134] or exert assistive force [135] to avoid obstacles or stay closer to an optimal path. Virtual fixtures can also be used to generate guiding forces [136, 137, 138, 139]. These methods often improve task-specific accuracies and execution times, however, they are also often associated with higher cognitive load and fatigue [140]. In order to mediate the issues described above, a number of works have proposed methods for modelling operators [141] and dynamically adjust the amount of assistance provided by the robot [142, 143].

1.5.5.3 Supervised control

Supervised control refers to a mode of operation in which the actions and movements of a robot are controlled primarily by the robot, with the operator able to provide input and supervision as needed. It allows the user to specify high-level goals or tasks for the robot to accomplish, while still taking advantage of the robot's onboard sensing and control capabilities and being able to intervene and provide supervision as needed.

According to Leeper et al., [144], conventional robot teleoperation control strategies that rely on autonomous modules performing certain aspects of the task tend to outperform those that require operators to handle more of the task independently. However, for these autonomous components to be effective, they must establish a sufficient level of trust with the operator and communicate their limitations clearly.

In supervised control the operator sets a number of goals for the robot to achieve, for example, consider works on navigation [145, 146, 147] or object grasping [148, 149, 150, 151, 152], and the robot performs necessary planning and executes the task.

The body of works on supervised control in VR-based robot teleoperation is relatively smaller than direct control - although supervised control is the primary control method in AR-based robot teleoperation and the works discussed in the corresponding section. It has been demonstrated that those methods lend themselves well to VR-based robot teleoperation, for example, Chow et al. [153] proposed a method for high-level supervised control of the da Vinci robot. Hetrick et al. [154] and Baker et al. [155] demonstrated that supervised VR-based waypoint control outperformed direct control in a number of robot teleoperation tasks.

An interesting control paradigm is to use direct teleoperation for tasks that robots have not seen before and use them to train corresponding autonomous machine-learning-

based algorithms [156, 157]. Zhang et al. [158] have demonstrated how VR-based robot teleoperation was used to collect a dataset of pixel-to-grasp demonstrations that were used to train a machine-learning algorithm for grasping.

1.5.5.4 Summary of robot control in VR

The summary of robot control options is presented in Table 1.2. Direct robot control poses several challenges such as real-time collision avoidance, self-collision and singularity avoidance, communication delays, and limited mobility in VR. However, direct control can be applied in scenarios with dynamic or unknown tasks, error recovery, and data collection for deep learning. An interaction proxy is recommended in VR-based robot teleoperation for a smoother experience. The challenges of shared control include the need for a structured remote environment that allows the robot to propose motion trajectories that can be blended with the operator's input, which is uncommon in VR-based robot teleoperation. Shared control can be used for operator training, and the operator can create virtual fixtures to provide guiding constraints. The supervised control method in VR-based robot teleoperation has certain notes and challenges, such as requiring a priori knowledge of remote environment and tasks for high-level commands, and direct control for error recovery. This method is applicable in scenarios with severe communication delays, where the task is known a priori, and for data collection for deep learning by demonstration.

Table 1.2: Robot control options: challenges and applications

mode	notes and challenges	applications
Direct control	• real-time collision avoidance with objects in remote environment, self-collision and singularity avoidance in joint space need to be considered carefully; • communications delay can cause teleoperation stability issues in bilateral teleoperation setups; • unilateral control is preferred in VR as it imposes less restrictions on operator's mobility in real world; • an interaction proxy is advised in VR-based robot teleoperation to decouple operator's and robot's movement for more comfortable teleoperation flow.	• teleoperation scenarios where the task is not known a priori or changes dynamically; • error recovery for supervised control mode; • data collection for deep learning by demonstration.
Shared control	• requires structured remote environment such that robot can propose motion trajectories that can be blended with operator's input; • uncommon in VR-based robot teleoperation as other two methods do the same.	• operator training; • virtual fixtures can be generated by the operator to provide guiding constraints.
Supervised control	• high level commands require some a-priori knowledge of remote environment and task; • error recovery requires direct control.	• robot control under severe communication delays; • scenarios where the task is known a priori (i.e. high level commands like grasping an object); • data collection for deep learning by demonstration.

1.5.6 Understanding human gaze in VR-based robot control

The human gaze can be used to control a robot directly, predict the operator's intentions such that robot can assist, estimate the operator's cognitive load, and give insight into the importance of objects and actions in robot teleoperation. Given that a considerable portion of the benefits of VR are visual, one can consider leveraging gaze tracking for the improvement of teleoperation.

1.5.6.1 Gaze saliency mapping

Human visual attention is a well-studied field that focuses on computational models that describe the human gaze, detect relevant ("salient") objects, and model gaze movements [159]. The goal of gaze saliency mapping is to understand how people process visual information and to identify the most prominent or attention-grabbing features in a scene. This information can be useful in a variety of contexts, including advertising, user interface design, and cognitive psychology research. Human gaze saliency is often modelled as a trade-off between bottom-up models, which use low-level visual features such as colour, texture, and contrast to predict gaze, and top-down models, which use higher-level information such as context and task-specific knowledge to make predictions [160, 161].

The majority of existing visual attention studies are done with a passive observer with static images, consider the body of work here to name but a few: [125, 162, 163, 140]; a few more recent studies utilised VR to perform visual attention studies with 360-videos [164, 165, 166, 167, 168]. A smaller number of works used scenarios where participants are engaged in some activity like driving [169, 170], gaming [171], human-robot interaction [172] or medical examination [173], or pick and place tasks [174]. Studies on visual attention in interactive VR scenarios are less common still: Hu et al. [175] proposed a method that used gaze temporal continuity to perform short-term gaze prediction in a VR-based game; Berton et al. [176] showed that there is little difference in gaze behaviour between the virtual world and the real during collision avoidance with a walker.

Gaze-based action prediction is a technique that involves using eye-tracking data to predict what action a person is likely to take next based on where they are looking. It can be used in a variety of contexts, including human-computer interaction, robot teleoperation, and virtual reality. There are several different approaches to gaze-based

action prediction, including machine learning algorithms that analyse gaze data and other contextual information to make predictions about the person's intentions. These algorithms can be trained on large datasets of gaze data and actions to learn patterns and make accurate predictions [177, 178, 179, 180].

1.5.6.2 Gaze tracking for robot control

The operator's gaze can be used to control a robot. Robot's desired position can be indicated as a single waypoint [181, 182, 183, 184] or a path [185, 186, 187] on a plane. Alternatively, gaze tracking can be used to press virtual buttons on either 2D screen [188, 189, 190, 191] or in VR, AR [192, 193, 194] to control the robot. Gaze tracking has also been used for dynamic camera control. [195, 196] proposed a method that puts whatever operator looks at the centre of the 2D display. In the context of 2D screen [197] concluded that despite the attractiveness of using gaze control for HCI and HRI, it is outperformed by conventional teleoperation control methods. Gaze tracking lends itself well to assistive shared control. [198, 199] proposed a method of gaze-based shared control for driving a wheelchair. One can also consider the body of work in [200, 201, 202, 203] that proposed methods for assistive and supervised grasping.

1.5.6.3 Gaze-based performance estimation

Operators' task load can be measured and correlated with teleoperation activities using: functional near infra-red stereoscopy [204], electroencephalography [205, 206] and gaze tracking [207, 208] to estimate the cognitive load; heart rate monitors as galvanic skin response to track stress [209, 210, 211]; electromyography [140, 212] and motion capture to track physical workload [213, 214]. Self-reporting questionnaires, like NASA-TLX [215] are also often used to estimate various aspects of task load [132, 216]. Given VR-based robot teleoperation's dependence on stereoscopic vision operator's gaze tracking becomes very interesting as a metric of the operator's attention and workload.

Gaze tracking is a promising area of research that has demonstrated its potential as an objective assessment tool for measuring performance and workload. Studies have shown that eye tracking can provide reliable quantitative data that can be used to identify task contributors to high workloads and assess the performance of individuals in various fields. For example, research has demonstrated that gaze tracking can be used to assess the performance of surgeons during training [217, 208], to identify workload

in drivers [218] and pilots [170], to evaluate the performance of nurses in interpreting bedside monitors [219].

In addition to its use as an assessment tool, gaze tracking has the potential to provide valuable insights into the cognitive processes and mental workload of individuals performing tasks. Eye tracking measures such as fixation duration, saccade frequency and duration, pupil diameter, and index of pupillary activity have been identified as important indicators of mental workload and can be used to identify factors that contribute to high workload in tasks [207]. By understanding the factors that contribute to high workload, it may be possible to design tasks and training programs that can help individuals perform at their best and improve overall performance. Overall, gaze tracking is a valuable tool for understanding and improving performance and workload in a variety of contexts.

1.5.6.4 Summary of gaze tracking in VR

Research in human visual attention aims to understand how people process visual information and identify the most prominent features in a scene. This information can be useful in various fields. Human gaze saliency is modelled as a trade-off between bottom-up and top-down models. Most visual attention studies are done with static images, but some recent studies use VR, scenarios where participants are engaged in activities, or interactive VR scenarios. These studies explore gaze behaviour, prediction, and short-term prediction in various contexts. Interestingly recent research in VR-based gaze tracking indicates that there is considerable overlap between VR and real world human visual priorities.

The operator's gaze can be used to control a robot's position or movement, either as a single waypoint or a path on a plane or by pressing virtual buttons on 2D screens or in VR/AR environments. Gaze tracking has also been used for dynamic camera control. Gaze tracking is useful for assistive shared control, such as driving a wheelchair or grasping objects. However, in the context of 2D screens, conventional teleoperation control methods outperform gaze control.

Gaze tracking is also a promising tool for assessing workload and performance, as it provides quantitative data to identify task contributors to high workloads and evaluate individuals' performance in various fields. Moreover, gaze tracking measures can indicate mental workload and help identify factors that contribute to high workload, leading to better task design and training programs. Therefore, gaze tracking is a valuable tool

for improving performance and workload in different contexts.

1.5.7 Literature review conclusions

While AR and stereoscopic vision have their applications in robot teleoperation, VR offers more flexibility in terms of how the remote environment is presented visually to the operator, as well as how the operator can explore and interact with the remote environment and the robot. For example, AR robotics applications are often limited to collocated robot interaction, as virtual elements need to be overlayed on top of the real robot, meanwhile, VR can be used to teleoperate a robot from an arbitrary distance. Stereoscopic teleoperation interfaces allow for the teleoperation of a robot from a set viewpoint, while VR allows the operator to teleoperate the robot from any viewpoint.

One of the key challenges in VR-based robot teleoperation is the reconstruction of the remote environment. Currently, this is typically achieved through the use of RGBD sensing and point clouds. However, point clouds can suffer from distortions and noise, resulting in less reliable visualisation of the remote environment. Additionally, point clouds do not effectively communicate objects' material properties.

The exploration of the operator's ability to manipulate the virtual world in VR-based robot teleoperation has thus far been limited to the generation of virtual fixtures to aid robot control and haptic feedback generation. However, as the operator in VR should have full control of the virtual world, there is a need for further research to explore novel VR-based interactions.

Another important aspect to consider in VR-based robot teleoperation is the operator's gaze. Prior research has demonstrated that gaze can be used as a control input for the robot and as a metric for participant performance and to infer human intention. Therefore, understanding the operator's gaze in VR-based robot teleoperation can lead to the development of more effective VR robot control interfaces.

2 VR-based robot teleoperation framework

Contents

Chapter summary: this chapter presents the VR-based robot teleoperation framework developed during the PhD and used in subsequent studies. It should be noted that the chapter presents the framework as hardware agnostically and specific hardware used in further studies is discussed in corresponding chapters. First, the requirements and design objectives of the framework are presented. Next, every subsystem of the framework is discussed in detail: leader-follower communications, capture, mapping, segmentation and visualisation of the robot and the remote environment, operator-VR interactions and navigation in VR, and robot control. The chapter concludes with a discussion of the limitations of the framework and potential improvements that can be made to it.

2.1 Framework overview

In order to perform studies in VR-based robot teleoperation a flexible experimental system was required with abilities to easily add/remove and adjust different functionalities on the fly like robot control modes, remote environment visualisation modes, human-robot interaction methods, etc. The ability to swap hardware such as the robot, the robot's sensors, the VR headset, and the operator's input devices and sensors was also desired. Therefore, a VR-based robot teleoperation framework was built that allows for all of the above and more. The framework is illustrated in Fig. 2.1 with nodes described in more detail in corresponding sections with the exception of material classification as it is specific to fibre optics-based tactile sensor, discussed in detail in section 3.3.

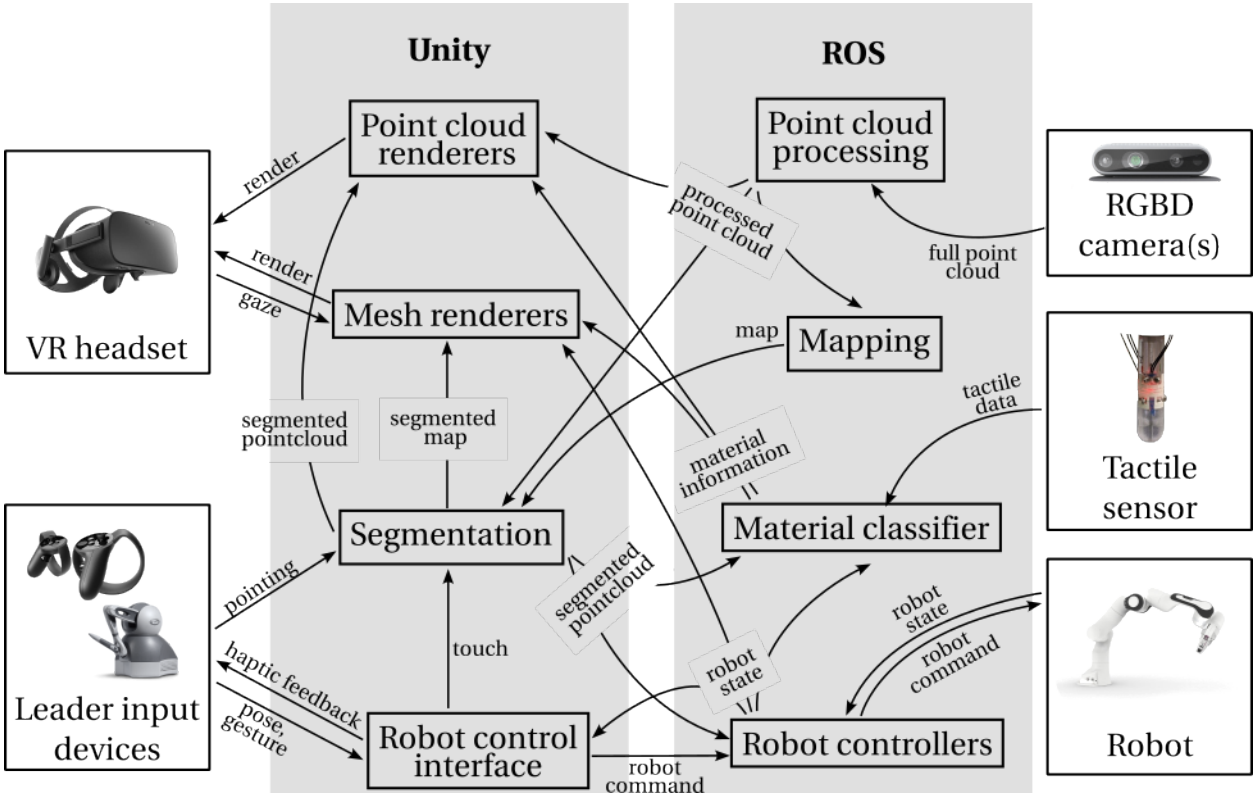


Figure 2.1: VR-based robot teleoperation framework scheme

In order to make the framework hardware agnostic the follower system is built with ROS - an open-source robotics middleware suite. Hence the framework supports ROS-compatible robots, direct and supervised robot controllers, ROS-compatible sensors like RGB and RGBD cameras and tactile sensors.

Most commercially available VR systems do not have Linux-based drivers making integration with ROS arduous at best. Therefore leader's VR interface is built with Unity - a game engine often used as a VR middleware supporting a wide range of commercial VR headsets and handheld controllers that can be deployed within the SteamVR framework. SteamVR framework unifies VR functionalities and button assignments across a large number of VR headsets and handheld controllers. Unity also supports a number of haptic devices. ROS-sharp and ROS-bridge were used to facilitate ROS-styled communications between the leader and the follower. Both the leader and the follower were on the same network, communicating over wireless or a wired network. It should be noted that ROS-sharp communicated with ROS at every visual frame update of the

VR-interface. All communications between ROS and Unity were done through the ROS-bridge except point cloud communications.

ROS-sharp cannot handle continuous streaming of point clouds due to the size of a single point cloud message. It is possible to communicate RGBD and depth as compressed images and perform pointcloud registration on the leader's interface in order to reduce the bandwidth. However, some RGBD camera drivers have registration algorithms that include other sensors, like built-in IMU in the point cloud registration process. In order to keep the communications uniform we chose to perform the point cloud registration on the follower's side such that all RGBD cameras can use their native point cloud registration methods and unload the point clouds from RGBD camera(s) to a dedicated UDP channel.

2.2 Visualisation, navigation, segmentation and mapping

2.2.1 Remote environment point cloud processing and visualisation

The remote environment was visualised as a point cloud collected from one or more RGBD cameras in the remote environment, for example, consider a configuration that uses an end-effector mounted dynamic "detail" camera and external static "overview" camera. In Unity, the point cloud is rendered using a particle system. Regardless of the RGBD camera(s) used the pointcloud was pre-processed before sending to the leader. If more than one camera is used, all point clouds are collected into a single point cloud message and all pointcloud coordinates are transformed from corresponding camera frames to the robot's base frame. This way in Unity any point cloud can be rendered from a single particle system with an origin at the robot's base as opposed to a separate particle system for each camera.

To reduce the bandwidth consumption the point cloud was cropped to the area of interest in front of the robot and all points on which registration failed were removed. The cropping box was exposed to the ROS dynamic reconfigure server such that it could be adjusted dynamically during teleoperation. Bandwidth requirements were further reduced by repacking limiting the point cloud to 15 bytes per point, as common RGBD cameras waste a considerable amount of memory: standard Kinect2 registration uses 32 bytes per point with 17 bytes as empty unused offsets and Intel d415i - 24 bytes with 9 bytes unused. Since the resultant pointcloud still exceeded the maximum UDP

packet size, the point cloud was sliced into smaller packets and sent with a corresponding packet number and a checksum. In Unity, the pointcloud was parsed in a dedicated parallel thread that was accessible by Unity's runtime to visualise the pointcloud.

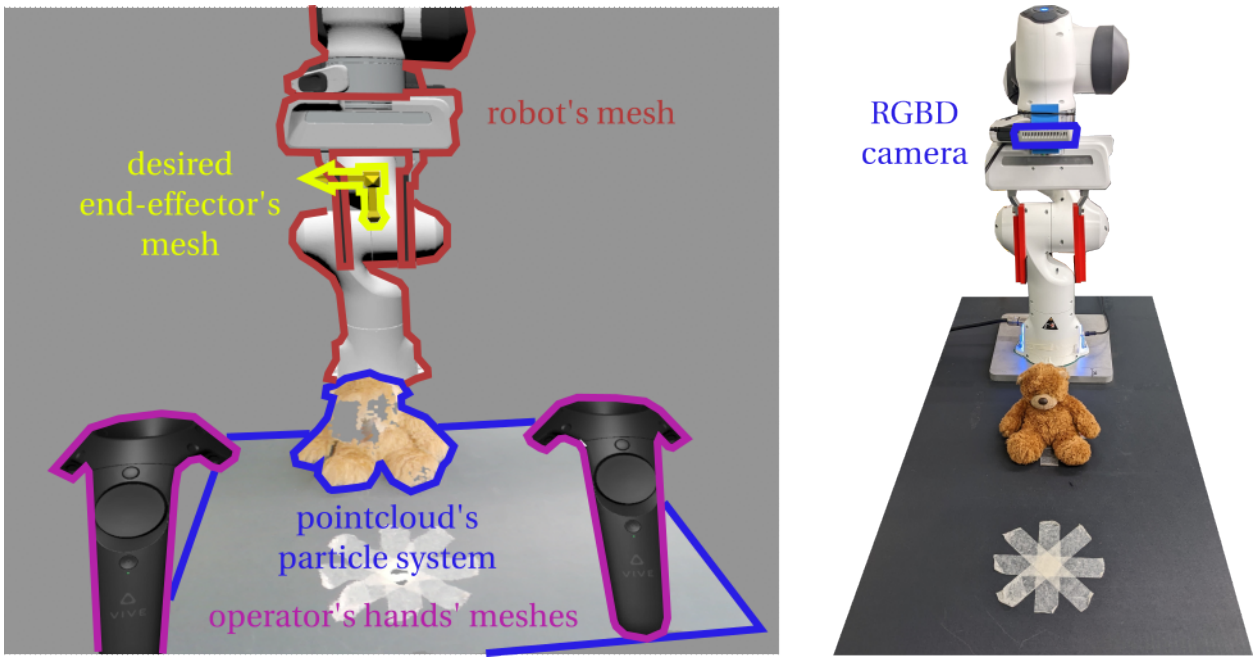


Figure 2.2: VR-interface and remote environment visualisation. Left: the operator's VR view (note that outlines and texts are not part of the VR view and only added here for clarity); right: the real remote environment.

The particle system is initialised with the maximum amount of points in the source point cloud, for example, two standard definition worth of points: 2 cameras $\times$ 640 widths $\times$ 480 height of points. The particle system's origin is set in the robot's base frame. On each visual update of Unity's VR runtime, the point cloud would update the positions and colours of particles from the parallel point cloud UDP thread. Since the UDP thread and particle system runs asynchronously, at each visual update particle system would be updated with the latest available data from the point cloud.

2.2.2 Visualisation of the robot and operator's tools

The robot is visualised in VR as an animated 3D mesh. The ROS-sharp suite allows importing Unified Robotics Description Format (URDF) of the robot to Unity. In default

real-time animation mode the robot's mesh was animated using the real robot's joint angle values. A secondary offline animation mode was used when the operator wanted to preview trajectories proposed by the motion planner. The offline animation mode can be toggled from the leader's Graphical User Interface (GUI). Once toggled the operator can preview the proposed trajectory one step at a time using an analogue stick on a handheld controller. The animation is switched back to the real-time mode robot when the operator rejected or accepted/executed the trajectory.

The framework uses an interaction proxy to control the robot - a 3D-axis mesh that represents the desired end-effector pose. We use different colours in order to indicate that desired end-effector mesh is idle or being manipulated by the operator. The operator's hands are visualised as 3D meshes styled after the real world controller or human hands. It should be noted that the operator's visual and motor axes are aligned as demonstrated in Fig. 2.2. The operator's GUI panel is a plane that displays the video stream from the RGBD camera, motion planning and grasping request buttons, logs from the VR interface, etc.

2.2.3 Remote environment mapping and segmentation

2.2.3.1 Mapping

End-effector-mounted cameras are usually closer to the area of interest than the external camera(s), therefore they have a smaller effective field of view. In order to compensate for the limited field of view of end-effector-mounted cameras, the remote environment is mapped using OctoMap [220]. OctoMap was also used as a simple point cloud meshing solution in our framework, as it produces bounding boxes around the point cloud that can be used as collision boundaries for Unity-based raycasting and collision interactions.

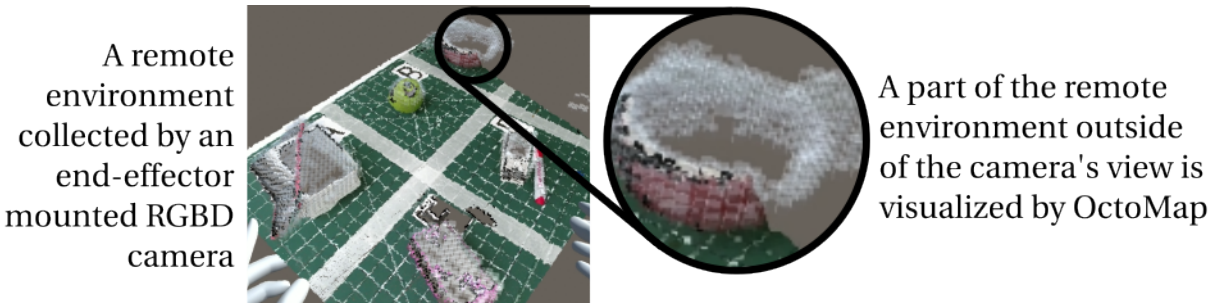


Figure 2.3: Remote environment mapping example

OctoMap is a widely used mapping solution that breaks down the space seen by an RBGD camera into nodes or voxels (virtual 3D cubes) of occupied/uncertain/unoccupied space, i.e. if space contains point cloud corresponding voxels are considered occupied, free space between the camera and occupied voxels is considered unoccupied and space occluded from the camera by occupied voxels is uncertain. Although more sophisticated mapping and meshing solutions do exist we chose OctoMap for its implementation simplicity and low bandwidth requirements. The OctoMap was produced with the same end-effector point cloud that was sent to the VR-interface after the point cloud was cropped to the area of interest (for example the area in front of the robot) but before the point cloud coordinate transformation to the robot's base frame (since point cloud coordinate transform would result in incorrect registration of OctoMap voxel types). The OctoMap was limited along the robot's base frame, for example, the floor/table on which the robot is standing such that the floor is not mapped but objects on it are, i.e. each object can have a separate bounding space. It is assumed that the operator can determine where the floor is themselves.

The standard ROS implementation of OctoMap was deployed and the map was sent to the VR interface in a binary format (other outputs of the OctoMap node - full OctoMap, point cloud voxel centres, and visualisation marker arrays - require considerably more bandwidth). A custom parser and renderer for the binary OctoMap was made, that inherits from the ROS-sharp subscriber class. The renderer only visualises occupied nodes with transparent boxes. We did not visualise or made a distinction between free and unknown nodes since the operator can infer that information themselves.

2.2.3.2 Segmentation

The point cloud and the OctoMap of the remote environment are segmented into separate objects such that different interactions with them can be performed, for example, autonomous grasp pose generation on a segmented point cloud of an object, assigning and visualising objects' material information, or registering operators' gaze.

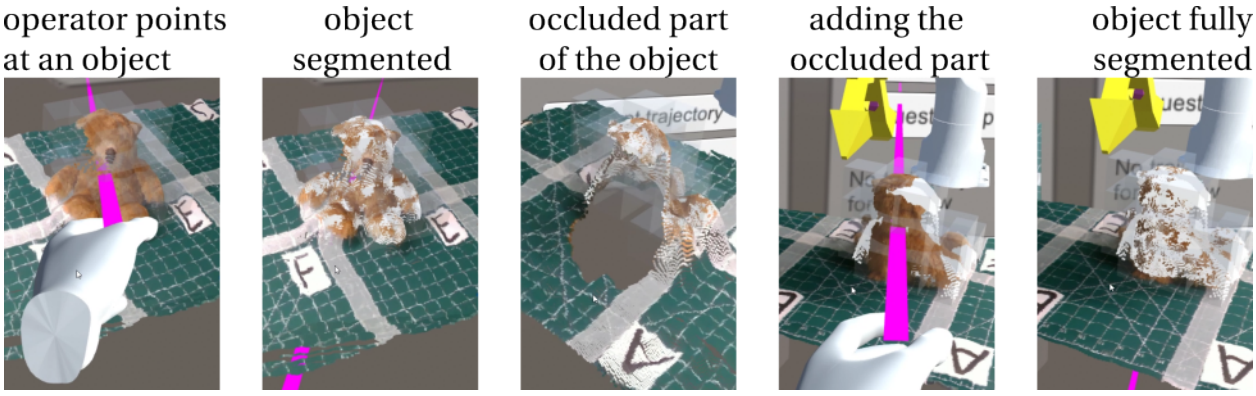


Figure 2.4: Object segmentation illustration

A very simple segmentation by separate occupied contiguous OctoMap nodes is used, illustrated in Fig. 2.4. Our OctoMap setup assumes that there is no clutter in the remote environment - i.e. point cloud of each object in the remote environment can be bound by a set of contiguous OctoMap voxels. The operator indicates the object of interest (the object operator wants to be segmented) by pointing at it with a virtual ray that extends from the operator's hand. The ray collides with one of the voxels that bound the object and "tags" it for the segmentation, and that voxel similarly further tags other connected voxels in up/down, left/right, forward/backward directions until all voxels that contain the object of interest are tagged for segmentation. All points contained in the isolated occupied contiguous OctoMap nodes as well as the nodes themselves are then segmented and cloned to persistent local memory. If the partial clone of the object is insufficient, for example, only one side of the object is seen by the RGBD camera the operator can reposition the camera and add more points to the existing clone. The segmented clone is visualised as a colour-coded point cloud to make it visually distinct from the real-time point cloud.

2.2.4 Gesture-based navigation

In VR-based robot teleoperation, the operator needs to be able to freely navigate the virtual space with minimal physical exertion, using minimum real world space, making as few real world rotations as possible, whilst being able to inspect any area in the virtual world, interacting with any object in the virtual world with high precision and have no motion sickness.

An alternative to the user moving in the VR space is moving the VR space itself around the user using hand gestures, similar to VR 3D drawing tools like Tilt Brush [221]. For example, instead of moving forward in the VR space, the user can use gestures to pull the world towards themselves. Although this mode of navigation is similar to joystick-based movement it does not induce motion sickness - the act of pulling an object toward themselves is more natural to humans than gliding forwards using a joystick. To the author's best knowledge, this navigation mode was only used by Thomason et al. [66] (contemporary of this work) in VR-based teleoperation before and lacks a common name, hence it will be called Gesture-based navigation (GBN) from now on. The gesture-based manipulation of the virtual world is illustrated in Fig.2.5. GBN can be used to navigate the virtual world whilst sitting, minimizing physical exertion compared to real world physical walking [98]. Gestures can be used to rotate the virtual world, such that the user doesn't have to rotate their torso and head compared to the teleportation navigation method [97]. Gestures can also be used to scale the virtual world up/down allowing the user to inspect the virtual in detail or have a broad overview. Furthermore scaling the virtual world down (zooming-out) allows the operator to control the robot using smaller hand movements further reducing the operator's physical load.

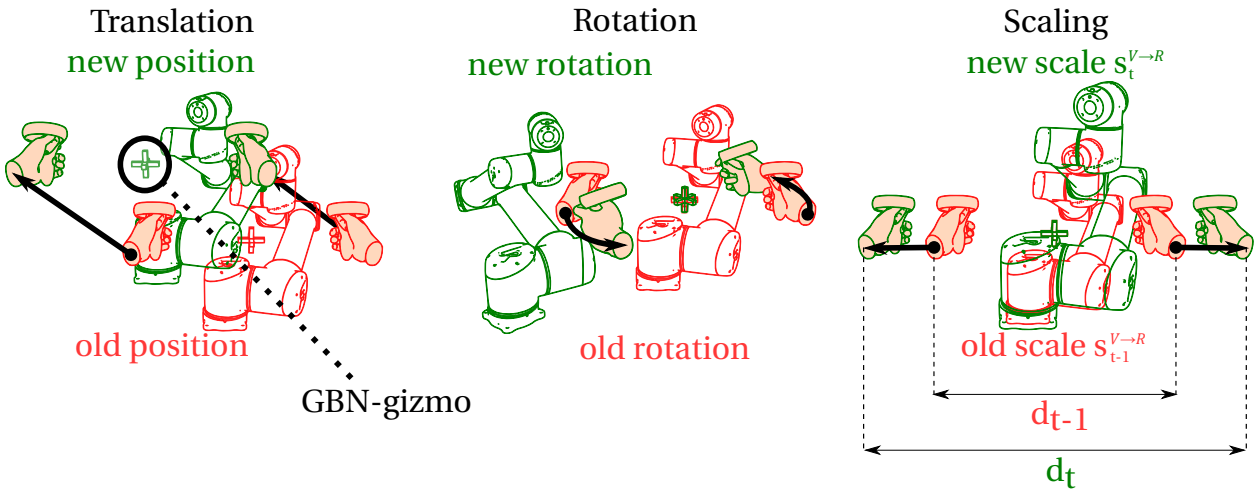


Figure 2.5: Gesture-based navigation

The GBN is engaged by pressing the gripper buttons on both handheld controllers. Gripper buttons are commonly placed under the middle and ring fingers of handheld controllers and are intuitively associated with grasping objects in the virtual world. Pressing both gripper buttons can then be intuitively associated with "grabbing" the virtual world. Once the virtual world is grabbed a GBN-gizmo is displayed between the user's hands. GBN-gizmo is named after "3D Gizmo" - a visualisation and control tool in 3D design applications that visualises objects transform in 3D space and allows translating, rotating and scaling of objects. In our implementation a GBN-gizmo is a 3D cross mesh which is used as the reference point - a virtual anchor point, for translating, rotating and scaling the virtual world.

In Unity implementation terms once the GBN is engaged all 3D objects in the virtual world (aside from the operator's head and hands) are set as children of the GBN-gizmo and any transform changes applied to the GBN-gizmo are applied to them as well. Once GBN is disengaged children are released. The GBN-gizmo is always placed on the midpoint of a vector that points from the operator's left hand to the right hand and is aligned along the vector.

To rotate the virtual world the operator should engage the GBN and perform a rotation as if rotating a physical steering wheel. The GBN-gizmo follows hands rotation and the virtual world follows. Similarly pulling and pushing the GBN-gizmo will pull and push all objects in the scene. The virtual world travels the same distance as the operator's hands, regardless of the scale of the virtual world.

While the translation and rotation can be performed using standard Unity tools, scaling requires a custom behaviour. Bringing hands closer/further apart scales the GBN-gizmo up/down and the virtual world follows. The scaling factor is notated as $s^{V \to R}$, which defines how objects in the virtual world are scaled compared to their real world counterparts:

$$w^R = s^{V \to R} w^V \tag{2.1}$$

with the w^R as the real world object size (e.g. the width of the robot base); w^V as the virtual world size of the same object. The scaling factor $s^{V \to R}$ is defined based on the operator's gestures as:

$$s_t^{V \to R} = s_{t-1}^{V \to R} + c \frac{d_t - d_{t-1}}{d_{t-1}} \tag{2.2}$$

with d_t as distance between handheld controllers at time t, c - constant gain. The scale is further clamped between certain minimum and maximum values to prevent accidents when handheld controller tracking is lost. Translation, rotation, and scaling gestures can operate simultaneously.

2.3 Robot control

The VR-interface is designed such that the robot control interactions would be the same across real-time and supervised control modes. In both cases, the operator controls the robot by setting a new desired end-effector pose in the VR interface. In the real-time mode, the robot continuously follows the desired end-effector position, while in the supervised mode, the trajectory is generated offline, previewed, and accepted/rejected by the operator.

In terms of Unity implementation, the operator needs to reach out with their dominant hand (by default the right hand) to the desired end-effector mesh until the collision meshes of the operator's hand and the desired end-effector intersect. At this point, the operator can press the trigger on the controller to grab the desired end-effector mesh. Once grabbed the desired end-effector is set as a child of the operator's hand and moves with it.

2.3.1 Direct control

In real-time control, the robot continuously follows the desired end-effector pose set by the operator. The desired end-effector pose is published similarly to Interoperable Teleoperation Protocol (ITP) [222] - as increment change (delta) of the desired end-effector pose. The change in pose is then applied to the target end-effector on the robot's controller. Real-time position-position and position-velocity (rate mode) controls are included in the framework. Both options can operate in unilateral (no force feedback) and bilateral (with spring-damper force feedback) modes similar to impedance controller [223].

2.3.1.1 Real-time robot control: position-position control

In position-position mode, the change in desired end-effector position is applied directly to the target end-effector pose on the robot's controller as illustrated in Fig. 2.6:

$$x_f = s^{V \to R} x_l, \tag{2.3}$$

x are pose vectors, subscripts f and l denote follower and leader variables respectively. The controller then solves the inverse kinematics problem to calculate new joint positions necessary to reach the target end-effector pose.

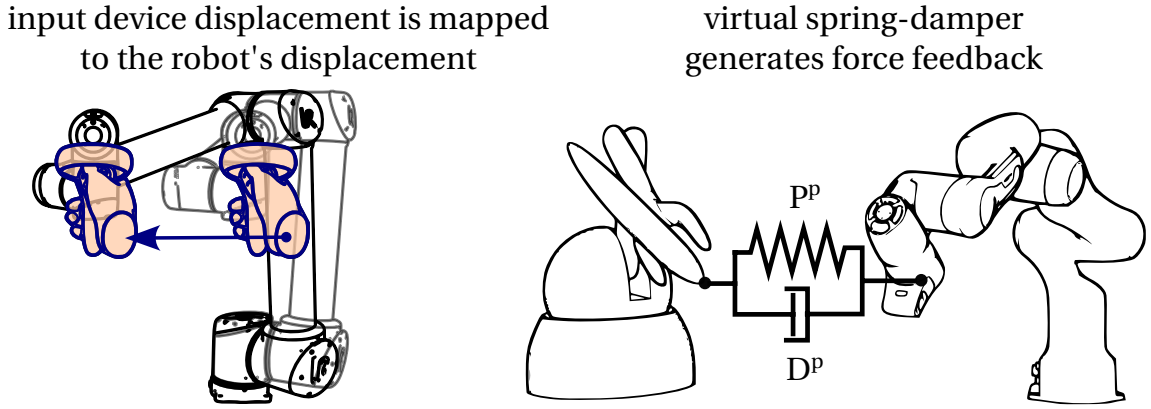


Figure 2.6: Direct position-position control mode. Change of pose of the leader device is mapped directly to change of pose of the follower robot. Top: unilateral direct position-position mode control using VR handheld controllers; bottom: bilateral direct position-position mode control using haptic device.

If a haptic device is used as the operator's input device then bilateral force feedback can be enabled on the haptic device and the robot by introducing a virtual spring damper system in between. Both the robot and the haptic device can render the corresponding force by applying a control law described by equation 2.4:

$$u_f^p = P_f^p(s^{V\to R}x_l - x_f) - D_f^p v_f, \tag{2.4}$$

where u is the control variable, P and D are stiffness and damping gains. Superscript p denotes the position-position tracking mode. The control law applies symmetrically to both leader and follower. Therefore in this particular case there is no difference between the leader and follower controllers, with exception of robot specific scaling factors and control gains.

Consider a case when the workspace of the haptic device is considerably smaller than that of the robot, but the task require high positioning accuracy that can be achieved with $s^{V\to R} \leq 1$. In continuous position-position mapping only limited part of the robot's workspace can be reached; in-order to cover more of the robot's workspace input device and robot need to be decoupled. A decoupled mode simply pauses the tracking between the leader and follower robots, allowing to reposition the leader robot to a more convenient pose. Further replacing absolute poses and tracking with corresponding change of poses, the control law then becomes:

$$u_f^p = P_f^p(s^{V\to R}\Delta x_l - (x_f - x_{f,\circ})) - D_f^p v, \tag{2.5}$$

Hence both robots try to match each other's displacement and the teleoperation can be initiated from any pose $x_\circ$. Naturally switching from decoupled mode to tracking will reinitialise the $x_\circ$.

2.3.1.2 Real-time robot control: rate mode control

In position-rate mode (rate mode) the change in desired end-effector position x_l is mapped to the robot's end-effector speed v_f as illustrated in Fig. 2.7:

$$v_f = k\mathsf{DZ}(x_l) \tag{2.6}$$

with k as rate mode scaling coefficient and the dead-zone function $\mathsf{DZ}(\cdot)$ that defines the area of the input's device workspace in which the value for the position of the input

device is assigned to be 0. The dead-zone for one-DoF displacement is defined as:

$$\mathrm{DZ}(x_l) = \begin{cases} -(x_l + r_{dz}) & \text{if } x_l \leq -r_{dz} \\ 0 & \text{if } -r_{dz} < x_l < r_{dz} \\ x_l + r_{dz} & \text{if } x_l \geq r_{dz} \end{cases} \tag{2.7}$$

with the size of the deadzone, r_{dz}. The equations above can be applied for positions and speeds of robots in Cartesian space and can be applied to single or multi-degrees-of-freedom robots.

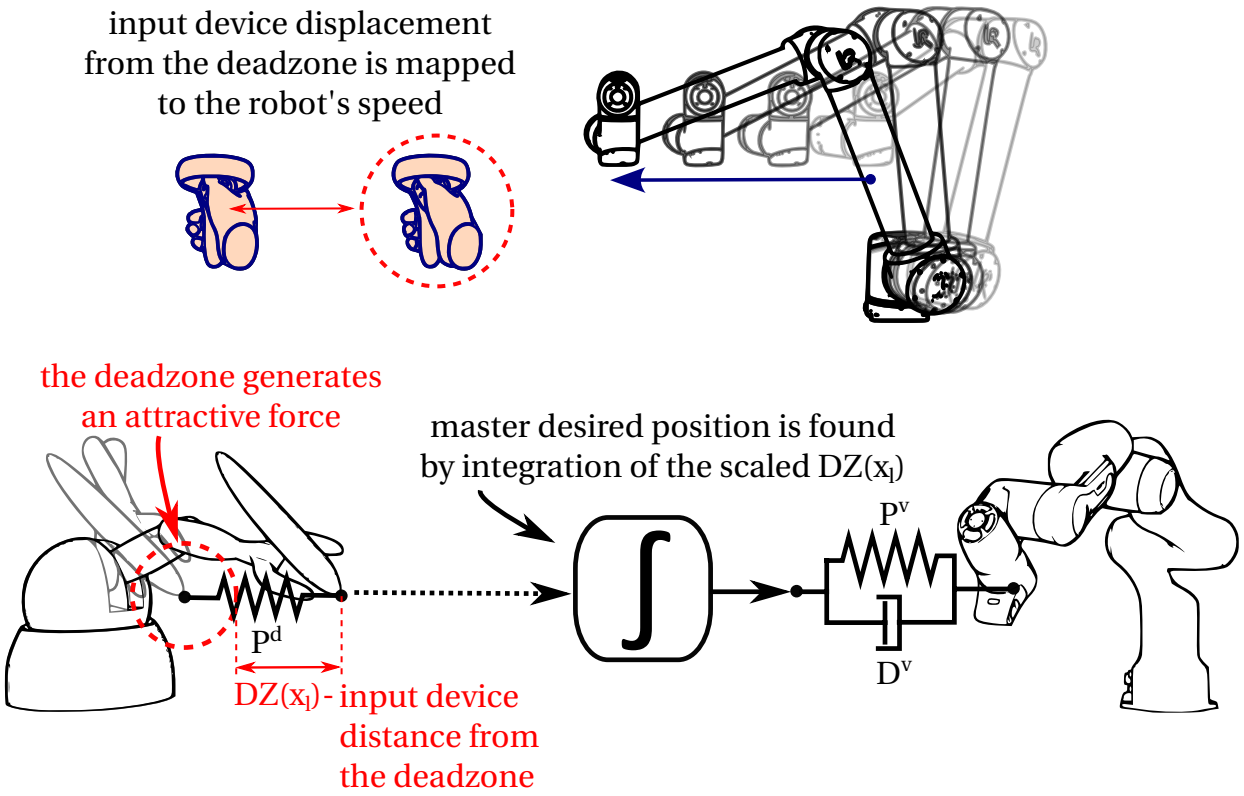


Figure 2.7: Direct position-rate control mode. Displacement from the deadzone of the leader device is mapped to the rate of the follower robot. Top: unilateral direct rate control using VR handheld controllers; bottom: bilateral direct rate control using haptic device.

In conventional teleoperation employing rate mode control, the gain k is constant. The framework introduces variable gain method that considers the scaling factor used in the VR representation of the remote robot and scene. Taking the proposed virtual

scene scaling into account, the rate mode control with variable mapping is following:

$$v_f = s^{V \to R} k \mathsf{DZ}(x_l) \tag{2.8}$$

The proposed variable rate mode mapping (2.8) allows for automatic position-to-speed gain adjustment based on the VR scene scaling. For example, if the operator scales the virtual representation of the remote robot and its environment to $s^{V \to R} = 2$ (i.e. the virtual world is twice as small as the real one), the calculated desired speed of the remote robot will be twice faster. In other words, instead of using the real world displacement of the master controller, the virtual world displacement of the controller is used.

If a haptic device is used as the operator's input device then bilateral force feedback can be enabled on the haptic device and the robot by introducing a virtual spring damper system in between. Since position-velocity mode is not symmetric different control laws apply for the follower and leader. The follower control law is fairly similar to one used on position-position mode:

$$u_f^v = P_f^v(s^{V \to R} \int k \mathsf{DZ}(x_l) dt - (x_f - x_{f,\circ})) - D_f^v v_s \tag{2.9}$$

As the integration is done on the leader, the follower still runs a position-position controller. Meanwhile the master control law is modified to include the deadzone spring:

$$u_l^v = P_l^v(s^{R \to V} \Delta x_f - (x_l - x_{l,\circ})) - D_l^v v_l - P^{dz} \mathsf{DZ}(x_l), \tag{2.10}$$

where P^{dz} is the deadzone spring stiffness.

2.3.2 Supervised control

In supervised robot control (also known as offline control) the robot's trajectory is first planned by a motion planner and inspected by the operator before the execution. After the desired end-effector pose is set, the operator requests a motion plan by publishing the desired end-effector pose. The pose is then used by a motion planner to generate a corresponding trajectory from the current state to the desired state. Based on the task and/or additional commands in the operator's GUI the motion planner distinguishes between simple move commands and task-specific commands (described below). The trajectory is published to the leader's VR interface for preview. After view-

ing the trajectory the operator can accept/reject the trajectory by publishing a corresponding boolean, triggered by a button press on a controller or leader's GUI. Optionally the trajectory previewing can be omitted such that the robot immediately executes the planned trajectory.

2.3.2.1 Task specific control: move-to-grasp-pose

The framework distinguishes between simple move and move-to-grasp-pose as the prior can result in trajectories that make the robot's end-effector collide and move the graspable object. The move-to-grasp pose adds an extra waypoint to the motion planner request such that the robot approaches the desired end-effector pose along the desired end-effector's z-axis (the z-axis is by default, reconfigurable based on the gripper used). The move-to-grasp-pose is published to a separate topic that can be triggered either from the operator's GUI or based on the virtual world context, for example assuming that graspable object is placed on the robot base frame's XY-plane (floor) any desired end-effector pose that is in the certain z-coordinate range are grasp poses.

Figure 2.8: Supervised control mode: move-to-grasp.

2.3.2.2 Task specific control: point-and-click grasping

Point-and-click grasping is an alternative to setting the grasp position manually. It uses autonomous grasp pose generation based on a graspable object's pointcloud. The operator points at the object that needs to be grasped and clicks a button on a handheld controller. The object's pointcloud is then segmented and published to the Principal Component Analysis (PCA)-based grasp pose generator. The PCA is used to estimate the object's orientation and centroid and principal axis, which are then used to generate the grasp pose similar to [224]. The generated pose is then treated as a move-to-grasp pose in supervised mode.

Figure 2.9: Supervised control mode: point-and-click grasping

2.3.2.3 Task specific control: tactile scan

The tactile scan is used to generate a sliding motion for the tactile exploration of objects. The desired end-effector pose set manually by the operator pose is used to generate the scan start and reverse/stop positions. By default the sliding motion is performed in a horizontal plane along the sensor's y-axis projection on the object's horizontal plane, assuming that the object is flat. The orientation of the tactile sensor with respect to the scanned object is determined by the orientation of the desired end-effector pose. By default, the robot approached and moved away from the scanned object along the sensor's z-axis.

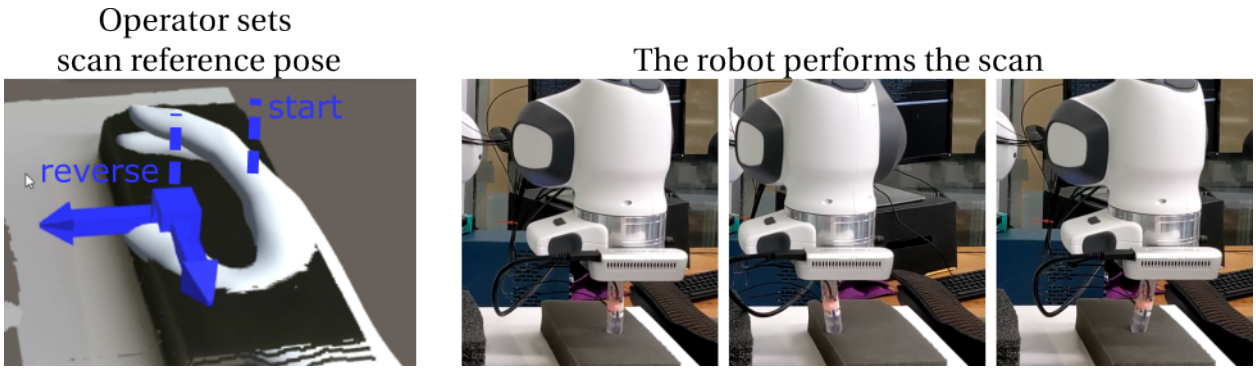


Figure 2.10: Supervised control mode: tactile scanning

2.4 Gaze tracking

Gaze tracking was performed using the Tobii Gaze-to-object-mapping (G2OM) machine learning model in Unity [225]. G2OM identifies the gaze vector and the most likely gaze candidate 3D object. This allows recording what object and where on the object the operator was looking at every VR interface frame update at an average 40Hz.

2.4.1 Tracking gaze on every object in VR-interface

An important gaze tracking pre-requisite is that every object in the VR-interface must have a collision mesh. Collision meshes are used for determining an object's bounds in 3D space and obtaining the gaze position in 3D space by performing a ray-cast from the user's headset along the gaze vector on the collision mesh. We ensured that G2OM can track the user's gaze on every 3D object present in the VR-interface by assigning them corresponding collision meshes and unique Unity IDs (tags). The robot, robot's end-effector, desired end-effector gizmo, and operator's right and left hands are represented in VR visually with high-resolution 3D meshes and have corresponding low-resolution collision meshes.

The rest of the remote background was visualised in VR using a Unity particle system based on a live RGBD camera's point cloud. The particle system does not have a bounding collision mesh by default. We used OctoMap as a rough point cloud (collision) meshing solution. Then OctoMap nodes could be segmented into intractable object(s) and remote background nodes (given that OctoMap was not truncated by z-height) by

individual nodes' z-coordinate in the robot's base frame.

The final element of the VR visualisation is the virtual space that is not occupied by any of the aforementioned objects and does not have its' own collision mesh. The virtual space was not bound (for example, the operator and the remote environment were not placed into a bound virtual room) to reduce the amount of gazeable objects. It is important to note that the gaze candidate and gaze vector may not correspond (readers are encouraged to consult [225] for details). This can occur if the operator looks at the empty VR space - G2OM assigns some other object as a gaze candidate. Hence it is assumed that whenever the gaze vector does not land on the gaze candidate the operator gaze is directed into empty VR space.

2.4.2 Gaze tracking calibration

G2OM SteamVR calibration was used to calibrate gaze tracking for operators. The calibration procedure is illustrated in Fig. 2.11: first the headset position on the operator's head is checked to ensure that internal headset cameras can view the user's eyes, then the interocular distance is set such that VR image is in focus, next operator is asked to track a dot on a screen, finally the eye tracking can be verified by checking whether gaze tracking correctly lights up the dots on the verification screen.

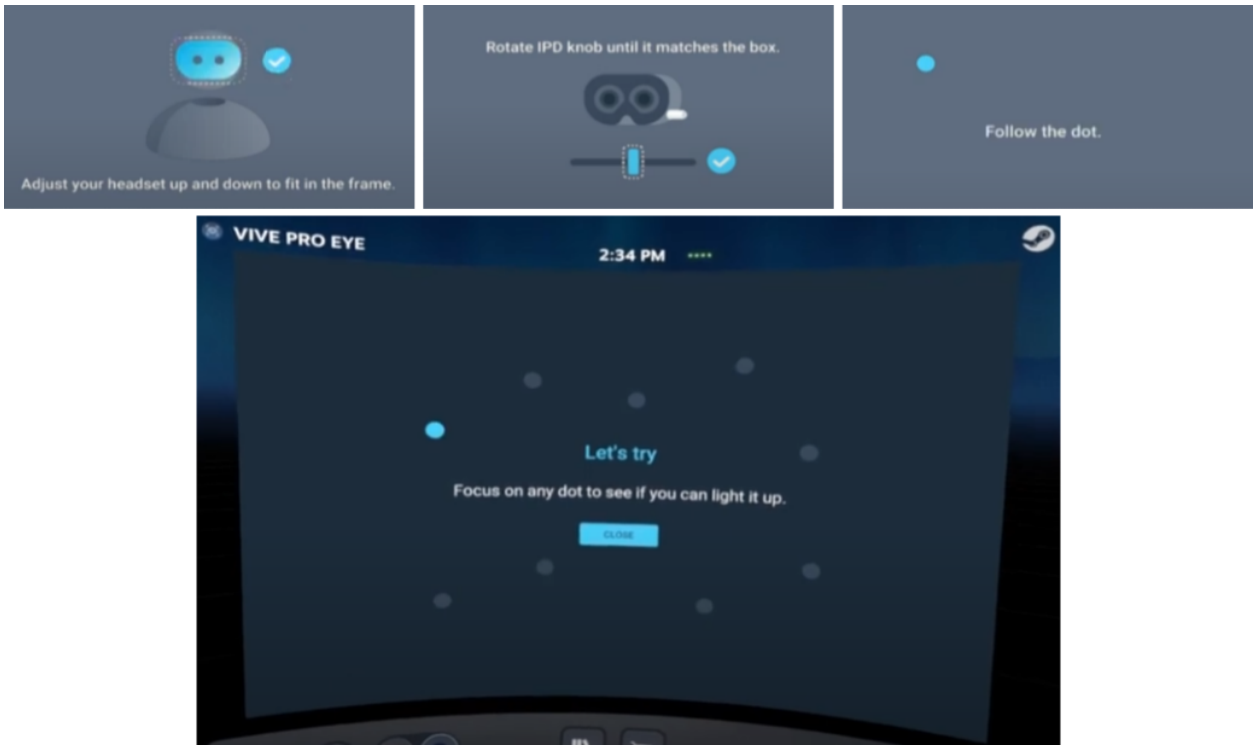


Figure 2.11: Gaze-to-object-mapping calibration

2.4.3 Gaze-to-object-mapping data

Tobii G2OM provides following eye-tracking data. A hierarchical list of the most likely gaze candidates, which helps to identify the objects that the user is looking at on the screen, note that confidence score is not provided. Gaze timestamps, which show when the user looked at different areas on the screen. This information can be used to analyse the user's attentional focus and engagement. Convergence distance and confidence boolean, which provide information about the user's depth perception and the level of certainty in their gaze behaviour. Gaze rays, including the origin and direction, for both eyes and the combined ray. This information can be used to analyse the user's gaze behaviour and visual attention. Blinking status per eye, which provides information about the user's eye movements and can be used to detect and analyse blinking behaviour. Pupil position and diameter as well as corresponding confidence booleans, which provide information about the size of the user's pupils and the level of certainty in the measurement.

2.5 Chapter conclusions

This chapter presented the VR-based robot teleoperation framework developed during my PhD and used in the study presented further. The framework is hardware agnostic, making it compatible with various robotic systems and cameras. Teleoperation and supervised control can be achieved with any ROS-compatible robot while visualising the environment through any ROS-compatible RGB and RGBD cameras. The framework also includes mapping and segmentation of the remote environment. Additionally, it allows for VR interface navigation and control through any Unity-compatible VR headset and controllers or haptic devices, with a consistent set of gestures and functionalities for operator ease of use. Furthermore, navigating the VR space is designed to be non-physically demanding and not inducing motion sickness, allowing for extended use.

In future work, the framework can be further extended by integrating additional ROS and/or Unity-based hardware, like stereoscreens, haptic devices or gloves and sensors. Another potential line of future work lies in improving mapping, meshing and segmentation of the remote environment, as currently, the framework uses the simplest to implement off-the-shelf solutions.

3 Remote environment visualisation in VR-based robot teleoperation

Contents

Chapter summary: This chapter presents two studies that focus on remote environment visualisation in VR-based robot teleoperation. In the first study, the operator's ability to understand the remote environment is investigated using four different visualisation modes: single external static RGBD camera, in-hand RGBD camera, in-hand and external static RGBD cameras, in-hand RGBD camera with OctoMap occupancy mapping. The results show that the latter option provided the operator with a better understanding of the remote environment whilst requiring a relatively small communication bandwidth. The second study proposes a method for tactile classification of remote environment objects' materials and their subsequent visualisation in the VR interface using tactile and proximity data as well as the robot's end-effector state feedback. Random forest, convolutional and multi-modal convolutional neural network classifiers were trained and compared. The results of material classification were successfully exploited for visualising the remote scene in the VR interface to provide more information to the operator.

3.1 Chapter introduction

This chapter delves into the difficulties of visualising unstructured remote environments in 3D using point clouds for VR-based robot teleoperation. Although point clouds are a primary method for visualisation, they often suffer from distortions, occlusions, and fail to represent objects' texture. Consequently, if objects in the remote environment are inaccurately represented in the VR reconstruction, it can lead to poor decision-making by the operator during teleoperation. Hence two studies are presented here. The first study investigated how the placement of RGBD cameras affected the operator's ability to visually explore a remote environment. The goal of the study was to determine such a visualisation configuration that would reduce the negative impact of point cloud distortions and occlusions on the operator's ability to understand the remote environment. The second study presented an innovative approach to material visualisation by

utilizing tactile exploration. The aim of this study was to determine objects' material and present them visually to the operator in a remote environment. Together the two studies significantly improve the remote environment visualisation in VR allowing the operator to make better decisions.

3.2 Dynamic field of view study

The majority of VR teleoperation systems use a single external static RGBD camera directed at the robot's workspace [4, 158, 130]. In such systems, the task performance can suffer from an incomplete and imperfect visual reconstruction of the remote environment. Depending on the remote environment's reflectivity, geometry, and overall lighting conditions some objects may appear distorted in the point cloud [69]. Occlusion is another issue limiting visual feedback in VR when a single camera is used - objects may be occluded by the robot or the clutter. Additionally, a single camera can only view one side of the object, hence only half of the object can be reconstructed in 3D.

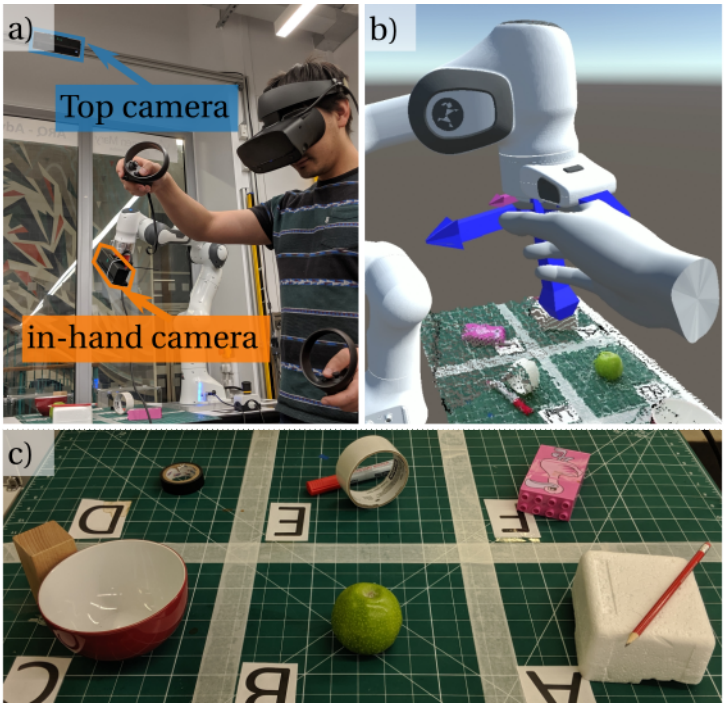


Figure 3.1: VR-based robot teleoperation with end-effector mounted camera

A possible approach to improve visual feedback in VR is to use an additional in-hand

RGB camera [4] (a video camera attached to the robot's end-effector (EE)) and render the video stream on a plane in VR. However, the grasp would have to be performed relying on the video stream rather than the point cloud which reduces the benefits of VR. Alternatively, the remote environment can be represented with the help of multiple external static RGBD cameras [3] that view the remote environment from different poses. Although it greatly reduces potential occlusions it does not eliminate them, and deploying multiple cameras in a remote site is not always practically feasible. Finally, there are methods that use knowledge of the remote environment and use existing mesh objects instead of point clouds [61, 62, 3]. These methods however have limited applicability in unstructured remote environments.

Aim of the study. In this study the effect of the RGBD camera placement on the operator's ability to understand the remote environment was investigated. The study hypothesis was that an in-hand RGBD camera combined with OctoMap [220] occupancy mapping provides the operator with a superior overview of the remote environment compared to the conventional single static camera [4] and multiple camera setups [3] for unstructured remote environments.

3.2.1 The experiment

3.2.1.1 Experimental task

The experimental task was to explore the remote environment in order to visually recognise and locate objects placed randomly in the remote environment. Four visualisation modes were implemented and used as experimental conditions (see table 3.1): **M1)** *external camera*; **M2)** *in-hand camera*; **M3)** *double camera* (external and in-hand); **M4)** *in-hand camera with OctoMap*. All modes used the same remote environment visualisation flow as described in section 2.2.1; in short: point clouds generated by RGBD camera(s) were cropped to the area of interest (a small area in front of the robot), coordinate transformed to the robot's base frame, and visualised in VR using Unity's particle system. In condition **M1** the RGBD camera was placed 2 meters above the robot's base frame as shown in Fig. 3.1.a). In **M2** the in-hand camera was attached to the robot's end-effector and pointed along the end-effector's z-axis. In **M4** point clouds of both the external camera and in-hand camera were combined into a single point cloud and visualised in VR in the same manner as **M1** and **M2**. The OctoMap was generated using the in-hand camera's point cloud after cropping but before point cloud coordinate

transform and visualised in VR as semi-transparent cubes (voxels).

Table 3.1: RGBD camera placement modes

mode	external (top) camera	in-hand camera	OctoMap
M1	yes	no	no
M2	no	yes	no
M3	yes	yes	no
M4	no	yes	yes

The remote environment in front of the robot was divided into a six-segment grid as shown in Fig. 3.1. Thirty-six different objects (varied in size, shape, colour, and reflectivity) were used for the visualisation study. Nine different objects were selected and placed in each of the grids for each trial (their locations and combination were randomised for each trial). Each object appeared for each participant once. In certain trials, some segments of the grid were left empty and some objects could overlap to create partial occlusions (clutter). The robot was placed into a random pose before each trial such that it would occlude part of the remote environment in modes that use an external static camera and/or only a section of the remote environment would be captured by the in-hand camera; in either case, the operator would need to move the robot in order to explore the remote environment.

Ten participants were recruited from Queen Mary University of London for the experimental study (all healthy adults; one female; age 25-30). All participants have signed the consent form in accordance with the ethics approval QMERC20.403. All participants had some experience with VR but did not have prior experience with robot teleoperation. Each participant was given a 10-minute training time during which participants got accustomed to the VR teleoperation interface and the testing procedure, using a separate training set of objects. Then participants performed the experimental task, once per experimental mode in a randomised order such that each participant would receive a unique permutation of experimental modes order (there were fewer participants than possible order permutations).

3.2.1.2 Experimental setup

The experimental setup was based on the VR-robot teleoperation framework presented in chapter 2. Here the specifics of the experimental setup configuration required for this study are discussed. The experimental setup consisted of Franka Emika's Panda robot,

two Microsoft Kinect2 RGBD cameras, Oculus Rift VR headset with Oculus Touch controllers, a leader PC, a follower PC and a local wired Ethernet network. The experimental setup is shown in Fig. 3.1.a,b.

The first Kinect2 camera was placed two meters above the robot. The second Kinect2 was attached to the robot's end-effector and pointed along it. Kinect2 point clouds were generated using the standard definition - 512×424 and cropped to the area of interest 0.9m×0.6m ×0.3m in front of the robot. The OctoMap representation of the environment was produced based on the in-hand camera's point cloud as described in section 2.2.3. The OctoMap resolution was set to 2cm. Only objects that were located above the desk were mapped. Only the occupied OctoMap nodes were rendered as semi-transparent cubes. Although the framework supports 2D RGB video streaming, it was disabled in this study as the study's focus was on the operator's ability to understand the remote environment from the point cloud reconstruction.

The robot was controlled by the operator in unilateral direct position-position teleoperation mode, as described in 2.3: the operator used Oculus Touch controllers to move the virtual desired end-effector mesh - blue axis mesh in Fig. 3.1.b), the pose change of the desired end-effector was then published to the robot's Cartesian Impedance based controller in ITP-protocol format and robot would match its end-effector pose to desired end-effector pose. The robot's real-time controller ran at 1kHz. Desired end-effector pose changes were sent from the leader to the robot's controller at every visual update of the VR-interface - 40Hz. The communication latency over a local wired network was considered negligible. The operator navigated the remote environment using GBN as described in section 2.2.4.

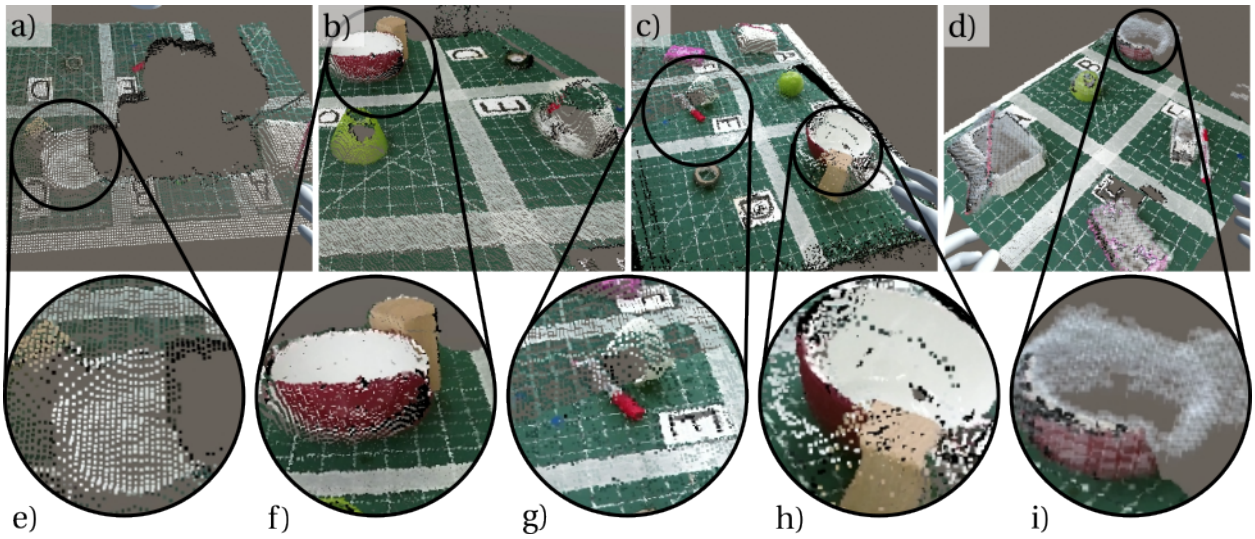


Figure 3.2: Operator's view in different modes: a,e) top camera: the robot occludes a part of the view and the bowl's shape is deformed; b,f) in-hand: the bowl is more recognizable; c,h) double camera: note the difference in bowl registration g) the edge between in-hand and top cameras - notice the difference in resolution; d,i) in-hand camera with OctoMap: the bowl is not in the camera's view but the OctoMap maintains an accurate geometrical representation.

3.2.1.3 Metrics

The task performance was characterised by task completion time, the number of correctly recognised objects and NASA-Task Load Index (TLX) [215]. In each trial, a participant was asked to identify an object type and its location (all grids were labelled) and then communicate verbally the object's name or colour and shape as well as its location. Participants had no prior knowledge of what object might appear in the grid. The number of incorrect object recognitions was counted. A full point was given for each incorrect answer. A half-point was given when the participant was able to locate the object but failed to recognise the object or the shape of the object; for example "green cone" instead of "green sphere" or "apple", see Fig. 3.2.b (an apple is located in the scene but the corresponding point cloud appears to look like a cone).

Participants had no prior knowledge of the total number of objects in the remote environment, nor were they informed if any objects were missed, hence participants could miss objects in clutter. After each trial participants filled in the NASA-TLX questionnaire. The average and peak communications bandwidth usage were recorded using

the number of points in the point cloud and the size of the OctoMap binary message.

3.2.2 Results

3.2.2.1 Visualisation and object recognition

The top plot of Fig. 3.3 shows that the visualisation based on the top camera **M1** was characterised by a high number of incorrect recognitions (ANOVA test, followed by pairwise T-test, $p<0.01$). From the experimenter's observations participants often failed to recognise the shape of an object if objects were non-convex or if the camera was looking at an object along its axis of symmetry, for example looking straight down at the bowl (see Fig. 3.2.e). Other objects were missed due to partial occlusions. In all other modes participants were able to correctly recognise the same objects thanks to the ability to direct the camera sideways as shown in Fig. 3.2.f,h. In double camera mode (**M3**), the mean number of misses was slightly higher compared to the in-hand camera only modes (**M2**, **M4**), although only statistically significant when compared to the in-hand camera with OctoMap mode **M4**. This is surprising given that more information can be accessed with a double camera. It could be caused by the imperfect overlap between the point clouds, see Fig. 3.2.h. If one of the cameras gives a false detection of the object's shape - the operator would not know which camera to trust. The in-hand camera with the OctoMap (**M4**) mode had the lowest number of incorrect object recognitions.

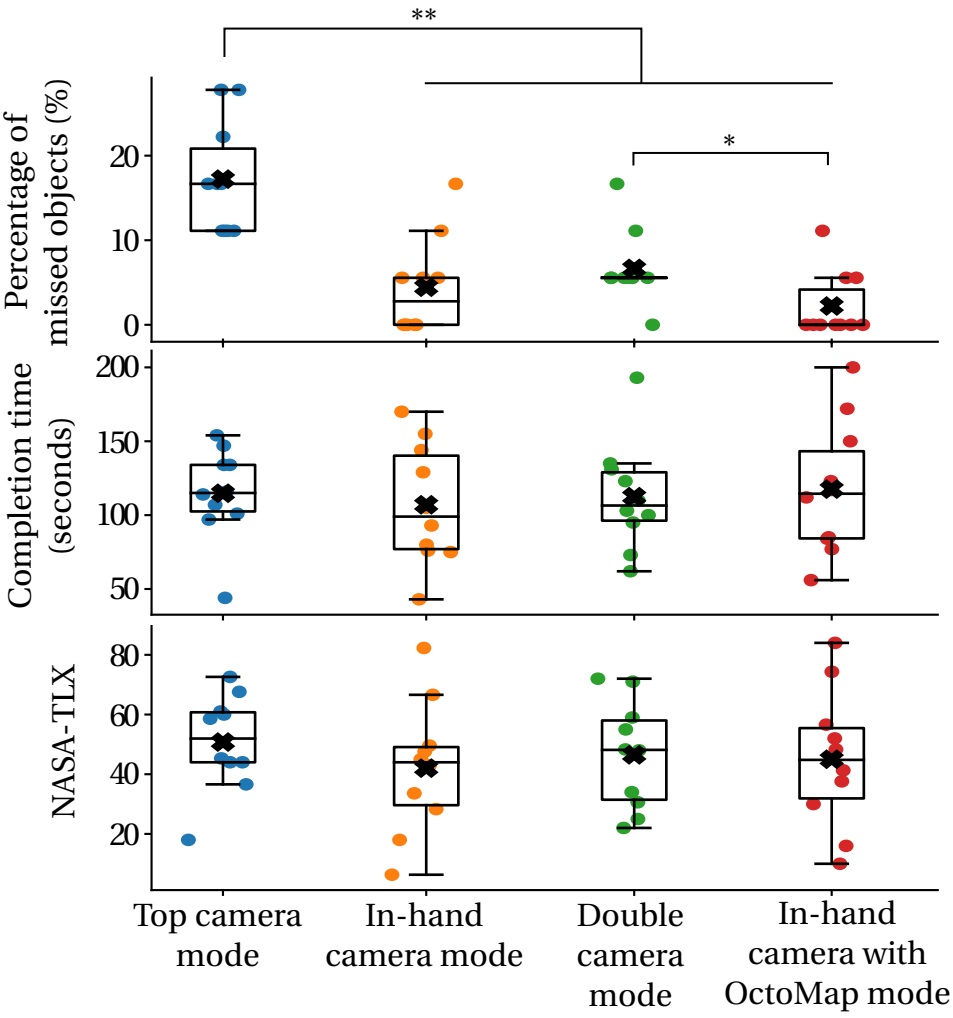


Figure 3.3: Boxplots of number of missed objects, task completion time and taskload. Boxes span test sets from the upper to the lower quartile with a line at the median, whiskers extend from boxes until the last datum within 1.5 interquartile range. Crosses mark mean values. Scatter points represent individual participants. *p $<$ 0.05, **p $<$ 0.01

3.2.2.2 Completion time and workload

The second and third plots of Fig. 3.3 show the results for task completion time and NASA-TLX scores across all participants per visualisation condition, respectively. The differences in completion times and NASA-TLX are statistically insignificant (ANOVA test). However, some interesting behaviours observed during the experiment can be outlined. Although completion times are comparable across modes, the distribution

of the time per task was different. In the in-hand camera mode, the participants spent more time re-positioning the camera; on the other hand, in the top camera mode, operators spent more time manipulating the view using gesture-based navigation. This is reflected by the shift from the mental demand to physical demand on the NASA-TLX breakdown, see Fig. 3.4 which presents the average NASA-TLX breakdown across all participants.

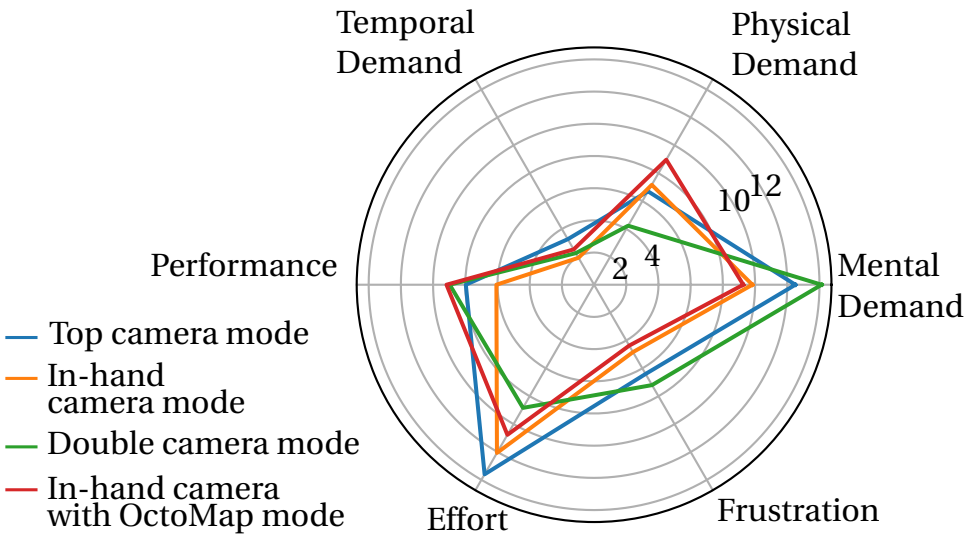


Figure 3.4: NASA-TLX breakdown

The double camera mode has the lowest effort value, which is to be expected given that it provided the most full representation of the robot's environment. The relative increase in the frustration and mental load was likely caused by the aforementioned imperfect overlapping between the cameras' views and reduced point cloud refresh rate (30Hz in double camera mode, 40Hz in single camera), caused by the asynchronous merging of top and in-hand point clouds.

3.2.2.3 Data transmission rate

The proposed VR visualisation modes rely on sending a significant amount of data from the follower site to the leader site. Therefore, the required communication bandwidth was compared between visualisation modes. The results are summarised in Table 3.2.

Table 3.2: Visualisation modes bandwidth comparison

Mode	Mean (MB/s)	Std.dev. (MB/s)	Max (MB/s)
Full PointCloud2	97.69	0	97.69
Top camera	7.78	0.78	8.82
In-hand camera	41.48	10.51	63.73
OctoMap binary	0.75	0.15	1.05

Since the top camera was placed much farther from the area of interest the portion of the point cloud that represents the area of interest is considerably smaller than the in-hand camera. As a result, the cropped point cloud of the area of interest is much lighter for the top camera compared to the in-hand camera. Naturally bringing the top camera closer to the area of interest would increase the point cloud resolution but it is arguable if it would exacerbate the underlying problems of point cloud occlusion - the robot would occlude more of the remote environment. The in-hand camera with OctoMap provides a comparable overview of the area of interest to double camera mode, but the OctoMap requires much less communication bandwidth than the top camera.

3.2.3 Discussion

The study showed that participants understand an unstructured remote environment better in an in-hand camera with OctoMap mode compared to static external (consider setups used in [4, 56, 158, 130]) and double camera modes (consider setups used in [72, 73, 71]). There also seem to be no obvious downsides to this visualisation mode as all experimental conditions had comparable task execution times and workloads. Furthermore, OctoMap provides an overview of the remote scene comparable to an extra camera but at a much lower communication bandwidth cost. Hence it can be concluded that for robot teleoperation scenarios that deal with unstructured remote environments, like [56] an in-hand RGBD camera with OctoMap is preferred.

There are a few study limitations that need to be discussed. First is the placement of the external camera, especially in mode **M1**. In this experiment the external camera was placed 2 meters above the robot's base frame which can be argued is further than similar setups, for example, [4] used an RGBD camera attached to the head of the Baxter robot which is considerably closer to the robot's arms. Placing the camera further resulted in a sparser point cloud which could have made identifying objects in the remote

environment harder for the participants. However, on the other hand, since the camera was further, the robot also occluded less of the remote environment (i.e. the robot cast a smaller "shadow" on the remote environment) and as a result operator had to move the robot less to view the remote environment. Hence one can argue that placing the RGBD camera closer would decrease the number of incorrectly identified objects but would increase task completion time and physical load.

Another important caveat that needs to be discussed is grasping. Grasping with an in-hand RGBD camera can be more difficult compared to grasping with an external camera since RGBD cameras require a minimal distance to an object for it to be registerable as a point cloud. In situations when the in-hand camera is too close to the gripper and the object, the operator has to grasp the object without or with limited visual feedback. Therefore three solutions for grasping were proposed: grasp by direct control, grasp on a pose, point and click described in section 2.3.

3.3 Object material classification study

RGBD cameras are widely used for reconstructing remote environments in VR-based robot teleoperation by generating point clouds [4, 3]. However, there are some issues with point cloud registration using RGBD cameras [69], for example, depth measurements may not work well on smooth metallic surfaces, which can cause noise in the point cloud registration process. Additionally, filtering algorithms can be computationally expensive [70], and point clouds may be distorted to the point that the original object is unrecognizable. Furthermore, point clouds do not provide much information about the texture and materials of objects.

Reliable information about the materials of objects in a remote environment is a critical aspect of teleoperation. Without this information, it can be challenging for the operator to perform tasks such as collecting all metallic objects in the environment. This difficulty is exacerbated by the fact that relying solely on the point cloud visual reconstruction of the environment can be unreliable, as demonstrated in Fig. 3.5, where three visually similar objects made of different materials are shown.

Machine learning is often used to classify materials from images using CNNs [82]. However, in low-light or hazardous scenarios [83], tactile exploration can be used to recognise materials through proximity, tactile, and force sensing [84, 85, 86]. Compliance properties can be investigated using hybrid force and proximity finger-shaped sen-

sors [87]. Fractures on surfaces can be detected using fibre optic-based sensors and a random forest classifier [88, 89, 90]. Deep learning models have been shown to outperform traditional machine learning algorithms by learning from high-dimensional raw data. Baishya et al.[93] found that a deep learning model for tactile material classification outperformed traditional machine learning classifiers, while Gao et al.[94] showed that combining visual and physical interaction signals with CNNs improved accuracy. Alameh et al. [95] converted 3D tactile data into 2D images for a CNN and achieved superior performance compared to traditional algorithms.

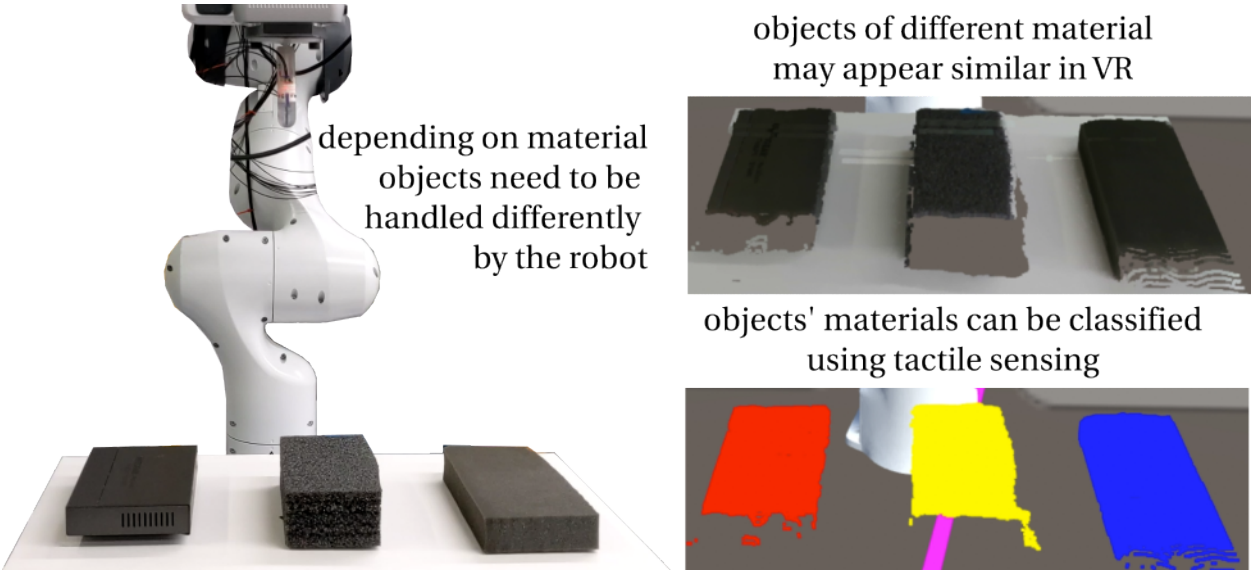


Figure 3.5: Object material visualisation in VR: objects of different material appear similar when rendered as point clouds in VR, which in turn can lead to operator taking incorrect decisions. Tactile sensing can be used to classify objects' materials and communicate them visually to the operator.

Aim of the study. This study proposes a method for tactile classification of materials for VR-based robot teleoperation - the operator remotely controls a robotic arm with a fibre optics tactile sensor to scan surfaces of objects in a remote environment. Tactile and proximity data as well as the robot's end-effector state feedback are used for the classification of objects' materials. Classification results are then used to visualise objects' materials in VR. Machine learning techniques such as random forest, convolutional neural and multi-modal convolutional neural networks were used for material classification. To the author's best knowledge, this work is the first attempt to demon-

strate the integration of material classification with tactile sensing in VR-based robot teleoperation.

3.3.1 Methodology

3.3.1.1 Experimental setup

Key components of the system are Franka Emika's Panda robot, a fibre optics tactile sensor (shown in Fig. 3.5), an Intel Realsense 435i RGBD camera, Oculus Rift S headset and Oculus Touch handheld controllers. Leader and follower computers were connected to a local wired Ethernet network. The robot was controlled by the operator using a supervised control as described in 2.3.2: the operator set desired motion path by manually placing waypoints in a VR reconstruction of the remote environment. These waypoints are then used to plan the robot's motion, which is previewed and accepted or rejected by the operator. The remote environment was visualised in VR using a point cloud from an Intel Realsense 435i RGB-D camera mounted on the robot's end-effector.

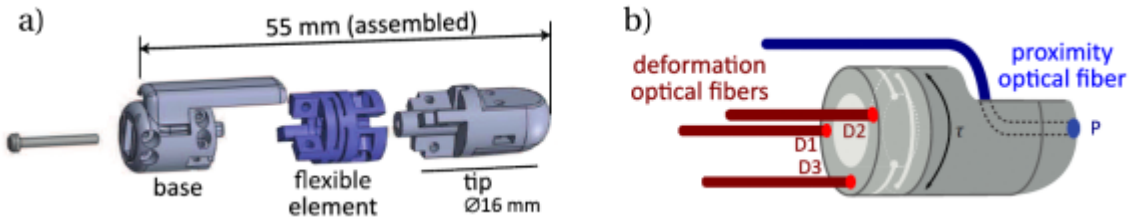


Figure 3.6: Fibre optics based tactile sensor: a) flexible 3D printed assembly b) deformation and proximity sensors

An integrated force and proximity finger-shaped sensor described in [87, 226] was attached to the robot's end-effector. The sensor shown in Fig. 3.6 consists of 3D-printed rigid and soft components that allow the finger to bend during interaction with the environment. The sensor has three pairs of optical fibre cables (D1, D2, D3) that use reflected light's intensity to measure the deformation of the finger. The fourth pair of optical fibre cables (P) is used to measure the distance between the tip of the finger and nearby objects. Each pair of the sensor's fibre optic cables was attached to a Keyence FS-N11MN light-to-voltage transducer that communicate with ROS at 400 Hz.

3.3.1.2 Tactile Data Collection

Supervised teleoperation was used for tactile exploration of five objects made of metal, paper, silicon, hard styrofoam, and soft foam. Fig. 3.5.a) shows sample objects made of metal, styrofoam and soft foam. All objects were flat and placed horizontally in the robot's workspace. An expert operator was asked to set the reference tactile scan pose, which determined the location and the orientation of initial touch between the end-effector mounted tactile sensor and the object as well as the subsequent sliding (scanning motion). The sliding (scanning) movement generation is described in section 2.3.2.3. The length of the sliding motion was set to 56.5mm performed in 8.3s. Training and validation data for the classifier were collected using supervised teleoperation such that the resulting dataset would be representative of real world use cases. Different orientations of the sensor and corresponding approaches have resulted in different sensor deformation behaviours. The contact force exerted by the robot/finger on the scanned object (and by the extent - the penetration depth on softer materials) depended on the reference pose height set by the operator. The operator set the reference pose and the robot performed the scan three separate times. Then the operator changed the reference pose by either changing the reference orientation, or reference position on the scanned object's surface or the reference scan depth. 150 samples (scans) per class were recorded.

Raw outputs of the tactile sensor were recorded for object classification. The robot's end-effector's average position error during the scan was recorded as the mean of differences between the end-effector's desired and actual positions. The robot's position error can be used as an indicator of objects' softnesses. For example, the tactile sensor can deform and push into soft foam resulting in a small position error, which is not the case for metal. Finally, the reference pose orientation was recorded as well. The tactile sensor scan output varied depending on the sensor's orientation with respect to the scanned objects, hence desired orientation was used as a feature in classifiers. A sample was recorded from 2 seconds before the slide start (to include the initial contact between the sensor and the scanned object) to 2 seconds after the slide end (to include the retraction of the sensor from the scanned object).

3.3.1.3 Data pre-processing for classification

Spectrograms of raw tactile data were used for objects' material classification. Spectrograms were generated with 52 time segments and 50 spectral bands (i.e. each channel's spectrogram is a 52×50 matrix). Three classifiers were compared: Random Forest (RF), Convolutional Neural Network (CNN) and multi-modal Multi-modal Convolutional Neural Network (MCNN). RF was made using scikit-learn library, CNNs were made using Keras with Tensorflow. Spectrograms were standardised using the maximum amplitude present in the dataset. The average end-effector position error was standardised using the maximum absolute value present in the dataset. The reference orientation was represented as a unit quaternion. Additional 50 synthetic samples were generated per class by randomly copying existing samples and adding corresponding $\pm 5\%$ standard deviation to each of the spectrograms' time segment and frequency band (to each cell of the spectrogram's 52×50 matrix).

The dataset was split into 60%-20%-20% training, validation and test sets, respectively. Samples were distributed into sets based on the tactile sensor's orientation with respect to the scanned objects. This was done to allow the classifiers to train with the data collected at different tactile exploration conditions. Fig. 3.7 demonstrates sensor orientations in radial angles present in the dataset (note that many samples overlap), where γ is the angle between the scanned object's z-axis (scanned surface normal) and the tactile sensor's z-axis and θ is the angle from scanned object's x-axis to y-axis (both are collinear to robot's base frame's x and y-axis). The training set contained samples in upper and lower 30% γ angles, validation and testing sets were randomly chosen from the rest. Hence classifiers were tested and validated on tactile exploration conditions that were not present in the training set, ensuring classifiers' robustness.

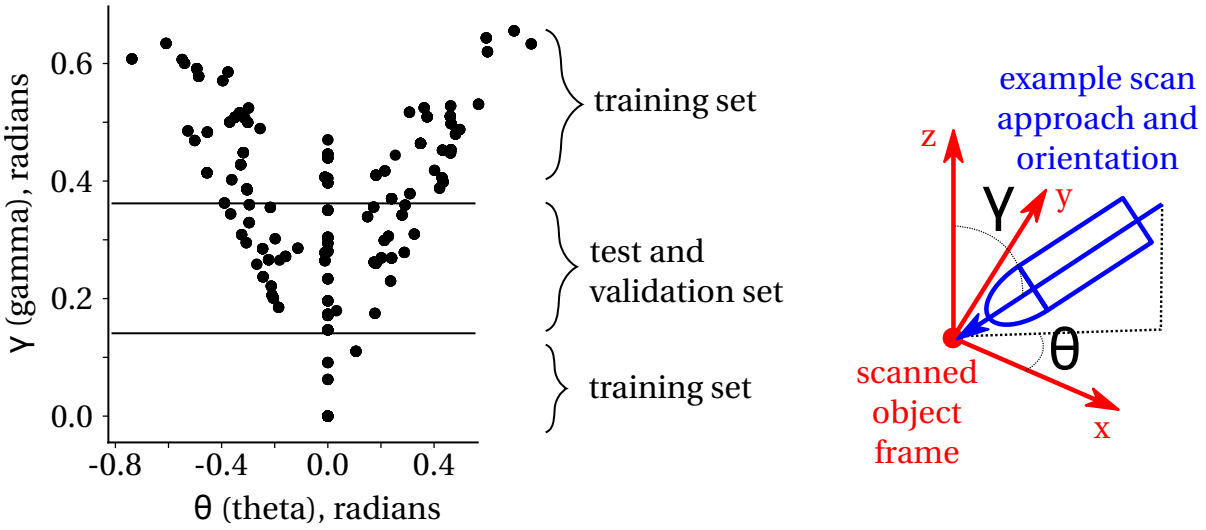


Figure 3.7: Distribution of different initial tactile scanning angular orientations of the sensor. γ is the angle between the objects' surface normal and the tactile sensor's z axis. θ is the angle between the projection of the tactile sensor's z-axis to the scanned object's xy-plane and the object's x-axis (object's x and y-axis are collinear with the robot's with an origin and desired scan position).

3.3.1.4 Classifiers

Classification with Random forest. RF [227] is a machine learning algorithm used for classification and regression built on an ensemble of multiple learning trees. Insensitive to over-fitting, it can produce reasonable predictions with a little tuning and provides an effective way of handling missing data. RF has been vastly used in remote sensing [228, 89, 88]. We implemented a Random Forest classifier with 1000 estimators. The number of estimators was determined by a grid search. The classifier was given concatenated spectrograms, average end-effector position error and reference orientation as inputs.

Classification with Convolutional Neural Network. CNNs are commonly used in image recognition due to their ability to learn cross-correlations between multiple channels (RGB in the case of images) and shift invariancy. They can also be used to learn patterns between multiple sensor signals as demonstrated in [229, 230]. We suggest a similar approach to classify objects' materials - using the tactile sensor's multiple sensing channels.

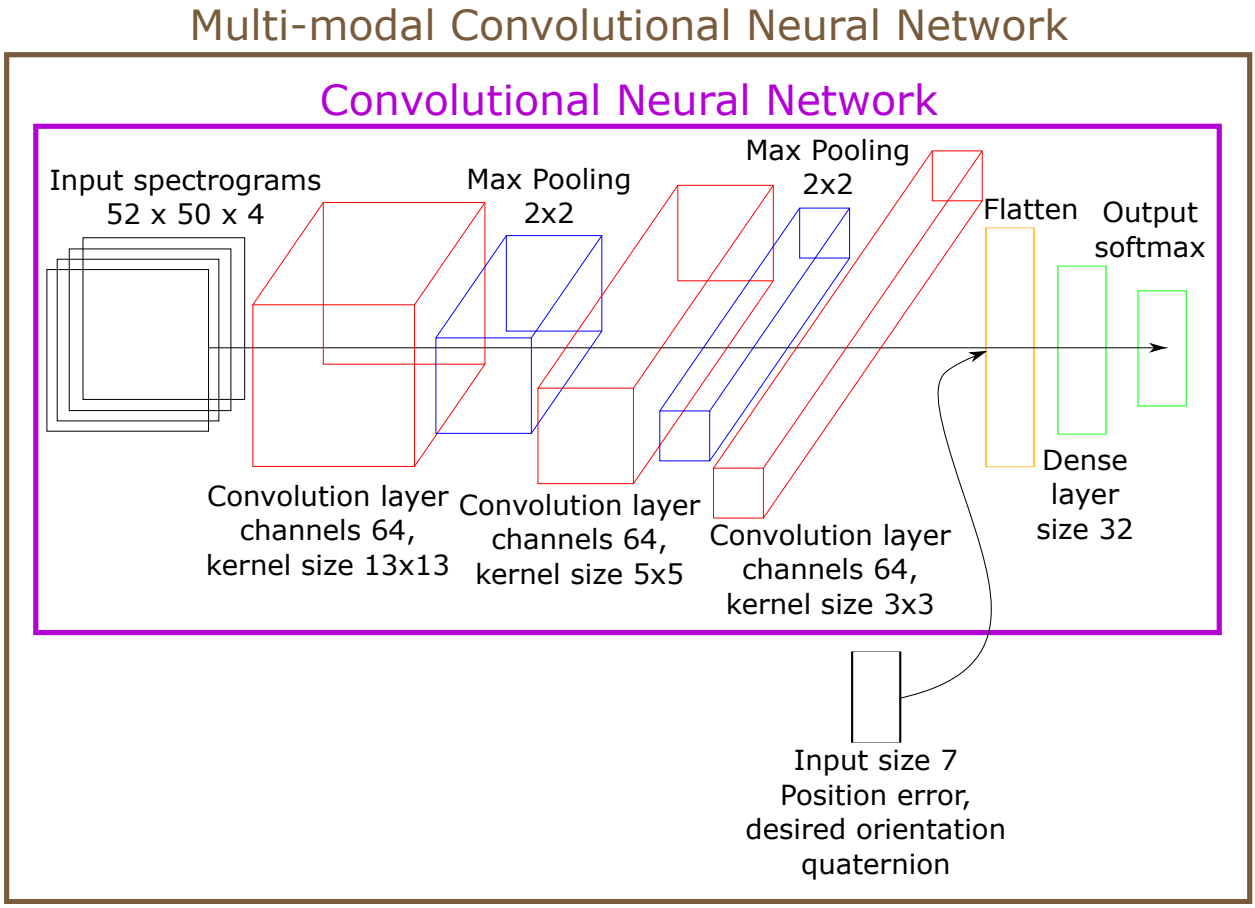


Figure 3.8: CNN and MCNN classifiers. Classifiers are similar except for the additional input layer that contains the robot's end-effector position error and scan orientation quaternion.

The CNN classifier only took spectrograms as an input and its' model architecture is shown in Fig. 3.8. There was a dropout with 30% probability between fully connected layers. The output of the last convolutional layer was batch normalised. The model was trained with an early stopping triggered by no improvement in validation loss. We used the Adam optimiser with an exponentially decaying learning rate. The model and training hyper-parameters were determined using grid search.

Classification with Multi-modal Convolutional neural network. An extra input was added to the CNN described above that included the robot's end-effector's average position error and the reference orientation. The model architecture is detailed in Fig. 3.8. The extra inputs were concatenated with the flattened output of the last convolution

layer. The multi-modal CNN retained the 30% probability dropout between fully connected layers and batch normalisation after the last convolutional layer. MCNN was trained with settings similar to the CNN.

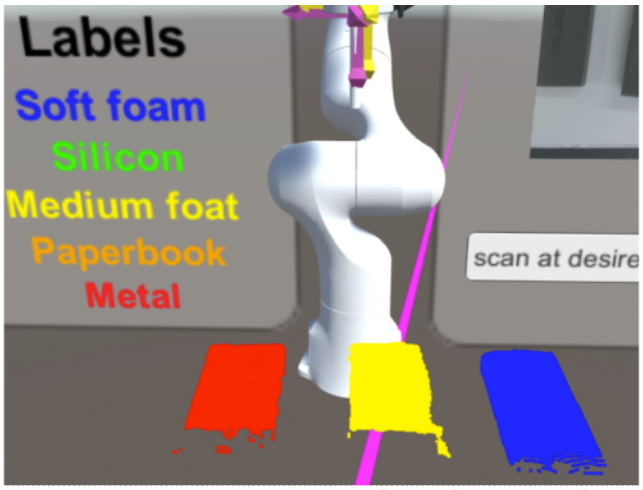
3.3.1.5 Visualisation of object's materials in VR

Two visualisation methods are proposed that communicate scanned objects' predicted classes to the operator in the VR-based robot teleoperation interface. Both methods rely on naive Octomap-based segmentation introduced in section 2.2.3, which allows the operator to segment point clouds into separate objects.

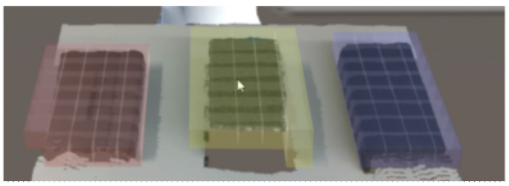
Objects' predicted classes as colour-coded Octomaps. As the operator sets the reference pose using the 3D axis mesh, the axis mesh checks for collisions with the Octomap using Unity's collider system. These collisions do not have any physical meaning and are simply used to detect which part of the Octomap the scan will be performed on. Once a collision is detected the corresponding Octomap cube and connected cubes are segmented and copied from the "live" Octomap and stored locally. Once the classification is finished the predicted class is used to colour the segmented Octomap according to the colour code given to the operator in the VR interface.

Objects' predicted classes as colour-coded point clouds. This visualisation method is similar to the one above except instead of segmenting and storing Octomaps, it segments and stores corresponding point clouds. Once the classification is finished the predicted class is used to colour the segmented pointcloud according to the labels' colour codes.

Fig. 3.9 shows the operator's VR view of classification results as coloured Octomaps and coloured point clouds as well as the GUI description of the colour coding. The operator can toggle between visualisation modes and "live" point cloud using a button press on a handheld controller. In the current implementation, both methods use static segmented clones of point cloud and Octomap respectively, which means that updates to "live" point clouds or Octomap of objects (for example if objects move or RGBD camera changes position with respect to objects) would not be reflected in classified clones. This presents a technical challenge that requires object tracking as well as continuous segmentation.



Predicted materials as
colour-coded segmented pointclouds



Predicted materials as
colour-coded OctoMap overlay

Figure 3.9: Classification results visualised in VR as static colour-coded segmented point cloud clones and as colour-coded Octomaps overlaid on corresponding objects. In both cases the corresponding colour code is shown in the GUI.

3.3.2 Results

3.3.2.1 Raw tactile data.

Fig. 3.10 presents three samples of raw sensor output for each material recorded at different sensor orientations and heights. There is a visually noticeable variance between different samples of each material. The leftmost vertical lines indicate the start of the slide, the middle one indicates the slide direction reversal, and the last vertical line indicates the stop of the slide.

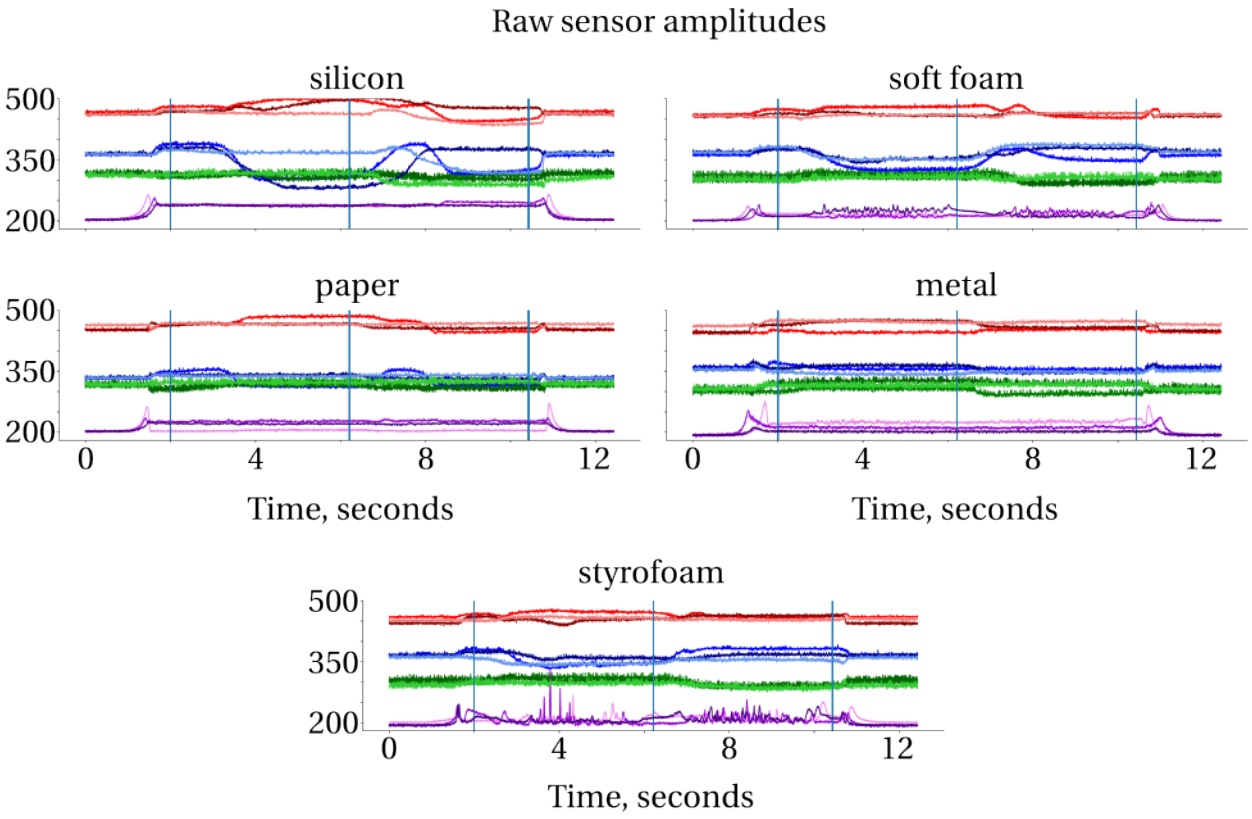


Figure 3.10: The raw output of the tactile sensor for different materials. Reds: D1, blues: D2, greens: D3, purples: P. Leftmost vertical blue line represents the start of the sliding motion; the middle vertical blue line represents the reverse in slide direction; the rightmost vertical blue line represents the end of the slide.

For all materials, the proximity sensor (P) spikes up on the sensor approach and retraction (before the slide begins and after it finishes respectively). The proximity sensor reading depends on the distance to the object. The proximity sensor only detects in a limited range and once the object is too close (or the sensor touches the object) the proximity sensor value drops to the baseline. The amplitude of the spike is also determined by the object's surface roughness, colour and reflectivity. In the case of rough and porous materials (soft foam, medium styrofoam), the proximity sensor generates a noisy output even during the touch.

There was a delay between the beginning of the slide and the deformation sensors' responses. The delay was the time necessary for the sensing finger to deform, (the sensor deformation can be seen in Fig 2.10. Similarly, responses of the deformation sensors

were delayed after slide direction reversal and sensor retraction.

3.3.2.2 Tactile data spectral analysis

Spectrograms were distinct per material, which was noticeable visually, see Fig. 3.11. Spectrograms differed most in the 0-15Hz frequency band, where there were large amplitude spikes during the sensor's initial contact with an object, during deformation and sensor retraction. There was also a noticeable high-frequency signal present in rougher/softer materials as well, (see D2 for soft foam).

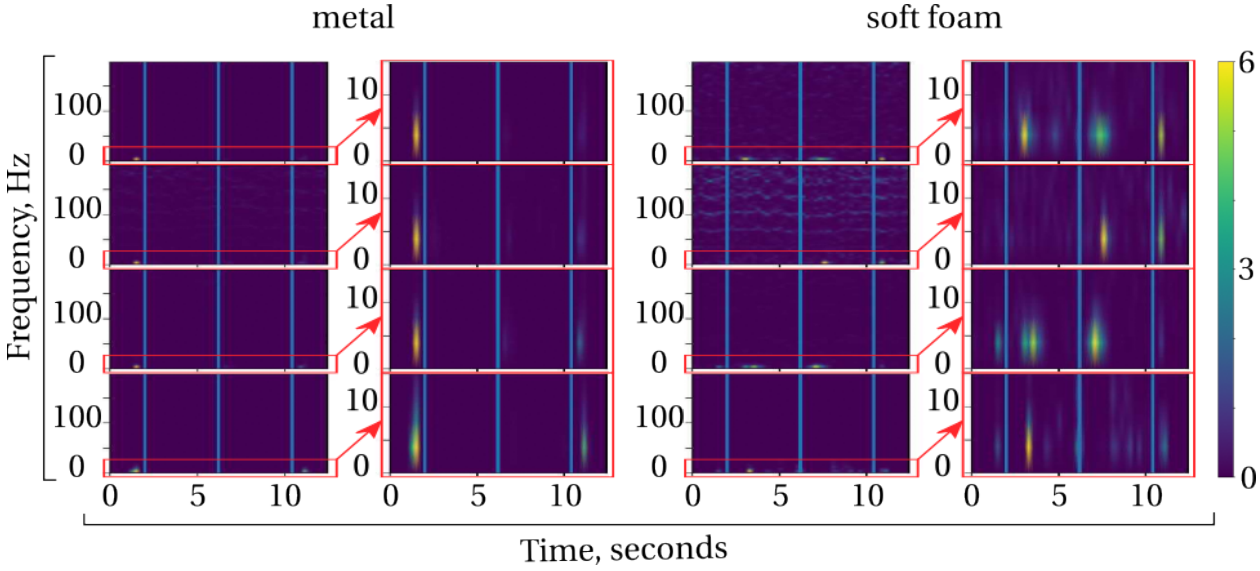


Figure 3.11: Sample spectrograms (not normalised) for metal and soft foam. There are large spikes in 0-15Hz frequency range for both materials (arrows indicate zoom-in regions) and smaller spikes in high frequency for softer/rougher soft foam.

3.3.2.3 End-effector position error

The average end-effector position errors occurred along the scanned surface's normal as the robot failed to push in as deep as the operator intended to. Table 3.3 presents means and standard deviations of the robot's end-effector's average position error along the scanned surface's normal per material. As expected the harder materials had larger errors: the sensor deformed and pushed in deep into the soft foam but could not do the same with metal. However, one-way ANOVA test showed no statistical significance.

Table 3.3: End-effector average position error on scanned materials

	Mean (mm)	**Standard deviation (mm)**
Soft foam	-0.17	0.18
Styrofoam	-0.12	0.15
Paperbook	-0.26	0.42
Silicon	-0.08	0.15
Metal	-0.23	0.22

3.3.2.4 Classification metrics

Accuracy, precision, recall, and f1-score classification metrics are used to validate and compare the used classification models. The results for the analysis with the implemented classifiers are presented in Table 3.4. Random Forest, which is robust to outliers and requires little parameter optimisation, achieved the best results. The MCNN, with the extra inputs of the robot's end-effector's average position error and reference orientation, achieves results comparable to the Random Forest classifier while the baseline CNN generates the worst outcome. The extra inputs make the MCNN model more robust than the CNN and able to generalise better.

Table 3.4: Classifiers' results comparison

Model	**Accuracy**	**Precision**	**Recall**	**F1**
Random Forest	94.5	95.0	94.5	94.4
CNN	84.5	85.1	84.5	84.1
M-CNN	92.0	92.3	92.0	92.1

Confusion matrices for CNN, M-CNN, and Random forest classifiers are shown in Fig. 3.12. One of the most challenging materials to classify for all three models is the paperstack, which is frequently confused with the silicon class, likely due to a similar pattern in the frequency domain.

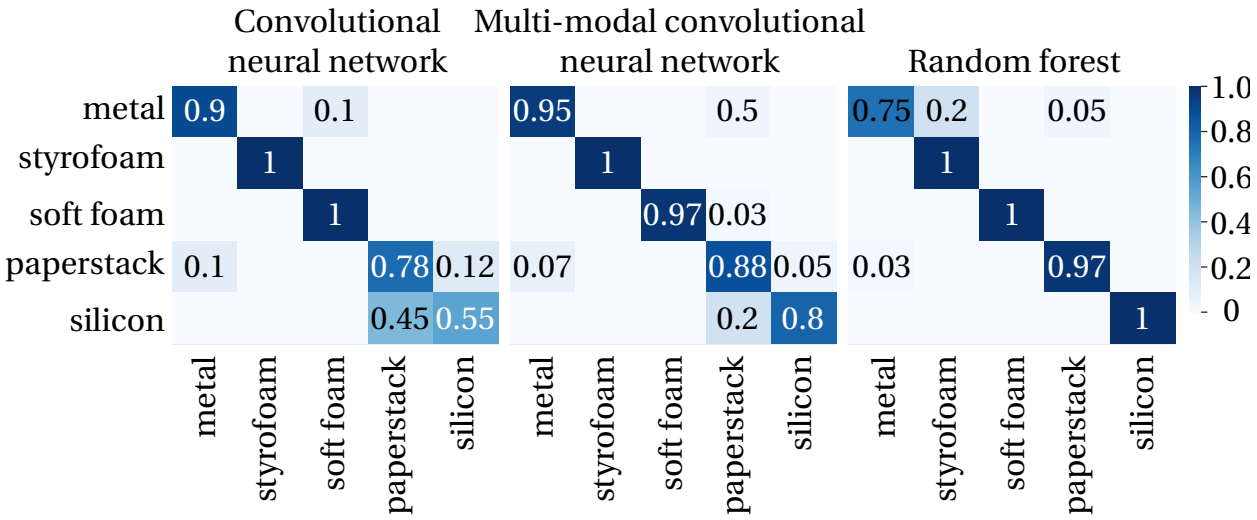


Figure 3.12: Confusion matrices of Random Forest, Convolutional Neural Network, Multi-modal Convolutional Neural Network classifiers.

3.3.3 Discussion

The results demonstrated that the random forest classifier (notably similar to [88, 89, 90]) had the highest accuracy at 94.5% followed by the multi-modal convolutional neural network (notably similar to [93, 95]) with 92%. Interestingly, a comparable accuracy of 91.7% was achieved in [93] using an electric skin-like tactile sensor (Grip VersaTek sensor 4256E) on a similar dataset. It should be noted that unlike in [93, 95] convolutional neural network did not outperform a more conventional random forest machine learning algorithm. However, given that the dataset was collected using constant linear scan length and duration, it is possible that the random forest classifier may not be able to generalise well to scans performed at higher or lower speeds. By comparison, it can be argued that MCNN would generalise better for scans at different lengths, speeds and curvatures. The results of material classification were successfully employed for visualising the remote scene in the VR interface to provide more information to an operator. It can be useful for teleoperation in hazardous environments like [83].

The dataset was collected using teleoperation to be representative of in-field operations. This however resulted in a relatively small dataset, a larger amount of samples would improve classifier training. The future work would benefit from larger sample sizes as well as from an extended object set. The latter would also allow moving from

hard-coded material-based labels to more general roughness/hardness estimation.

The proposed method currently works only with flat surfaces. In the case of non-flat surfaces, a similar approach can be used where the operator sets multiple waypoints that can be used to generate a non-flat trajectory. Alternatively, the scan path can be procedurally generated if given an accurate point cloud meshing solution.

Due to the imprecise nature of point clouds, the operator may put the reference pose too deep into the object potentially damaging the sensor or not deep enough, resulting in poor classification. In future work, one can focus on the use of the proximity sensor and haptic feedback to place the sensor more accurately at the object's surface. Proposed methods require a full scan to be completed before the object's class is determined. A potentially interesting follow-up topic is real-time object materials classification using Long Short-Term Memory networks.

3.4 Chapter conclusions

This chapter focused on the challenges of remote environment visualisation as point clouds for VR-based robot teleoperation. Point clouds are the primary means of visualizing unstructured remote environments in 3D. However, in the current state of technologies point clouds often suffer from distortions, occlusions and do little to represent objects' texture. If objects in the remote environment are inaccurately represented in the VR reconstruction, the operator can make incorrect judgments about their nature and/or shape, leading to poor decision-making during teleoperation. This chapter presented two studies: the first examined RGBD camera placement effect on the operator's ability to visually explore the remote environment; the second presented a novel method for visualising objects' materials using tactile exploration.

The visual exploration participant study has shown that an end-effector mounted RGBD camera with OctoMap mapping of the remote environment allows the operator to explore the remote environment with less point cloud distortions and occlusions whilst using a relatively small bandwidth. Admittedly, point cloud distortions that are often seen in teleoperation setups that use a single static external RGBD camera are a technical challenge, and while advancements in RGBD sensing technology can be expected in the future, researchers must currently consider camera placement carefully. Additionally, the study highlights the potential benefits and challenges of using a dynamic camera. Continuously repositioning the camera for a better view of the remote

environment can be cognitively and physically demanding. To alleviate this problem, automating the camera movement based on the operator's motion capture and visual attention data could be a viable solution. Furthermore, the study has shown that remote environment mapping should be given more attention and consideration. In this study, the OctoMap was used for its simplicity, but more complex meshing approaches that can represent the remote environment with greater accuracy could be considered - for example consider replacing point clouds of recognised objects with corresponding meshes [62]. Finally, it needs to be noted that grasping objects with an RGBD camera attached near the gripper can be problematic due to the camera's minimum distance requirement for point cloud registration. Various grasping methods to alleviate this issue are proposed in section 2.3.

The tactile exploration study aimed to further address the challenges of point cloud visualisation by providing the operator with information about the objects' materials. A novel method for presenting this information visually in VR-interface was proposed in order to increase the information available to the operator of the remote environment, potentially leading to improvements in the operator's decision-making.

The study was conducted using a fibre-optic-based tactile sensor designed for nuclear environments, for which no existing object material classifier existed. Therefore, novel machine learning-based classifiers were developed, including a random forest classifier, a convolutional neural network (CNN), and a multi-modal CNN. Classification results showed that the random forest classifier had the highest accuracy, but it is acknowledged that the multi-modal CNN may generalise better for tactile scans of different lengths.

The study also investigated the use of human-based data collection to train the classifiers such that training data would be similar to real world use cases. In retrospect, more accurate results may have been achieved through automated data collection, as it would have allowed for a larger number of samples to be collected.

In future studies, more attention could be paid to better segmentation and meshing of the remote environment as well as proximity sensor utilisation such that tactile scan path generation can be automated. Furthermore, one can consider moving from post-scan classification to real-time classification by using Long Short Term Memory networks.

4 Workspace scaling and rate mode control for VR-based robot teleoperation

Contents

Chapter summary: this chapter presents a study that investigates the effect of rate mode control with constant and variable mapping of the operator's joystick position to the speed (rate) of the robot's end-effector. The variable mapping depended on the virtual world scale. The study demonstrates how the rate mode control and variable scaling based on the VR reconstruction scale can be efficiently used for seated VR-based

robot teleoperation when the operator's arms are supported to reduce tiredness. The corresponding participant study shows that variable mapping allowed participants to teleoperate the robot more effectively, by adjusting the VR visual scale albeit at a cost of increased perceived workload.

4.1 Chapter introduction

Typical approaches in VR-based teleoperation interfaces include a direct position-position control which maps an operator's handheld controller's motion directly to a remotely controlled robot-manipulator [99, 2, 4], or a higher-level semi-autonomous supervised control which can be used by the operator to define the reference positions, goals and tasks for a robot to implement autonomously [47, 48, 46].

In the context of VR-based robot teleoperation that uses handheld controllers as input device rate mode control can be used to reduce the operator's fatigue. Consider Fig. 4.1.a) where an operator is using full arm movements to control a robot in position-position mapping mode. Teleoperating a robot using full arm movements can be fatiguing over long teleoperation sessions. By contrast, rate mode control allows the operator to use relatively smaller movements, for example only elbow and wrist movements to teleoperate a robot while the whole arm is supported as shown in Fig. 4.1.b) potentially reducing the metabolic costs and fatigue [231].

The advantages of virtual world dynamic scaling in direct position-position control modes are fairly clear - the operator can scale the world down to perform large movements and scale the world up to perform precise movements. By contrast, the advantages and potential disadvantages of virtual world scaling in rate mode control are far less obvious.

In rate mode control, the displacement of the input device from the input device's deadzone is mapped to the desired speed (rate) of the remotely controlled robot, see Fig 4.2.a). Rate mode control is practical for telerobotics applications in which the input device's workspace is significantly smaller than that of the remote robot, e.g. mobile robot teleoperation, where joystick displacement is mapped to a mobile robot's speed (similar to a car's gas pedal) [232, 233, 234], or robot-manipulator control - when mapping the movement of a small haptic device onto a robotic arm [235, 9].

It is unclear how visual scaling of the virtual representation of the remote robot will affect the performance of the operator when rate mode control is used. In other

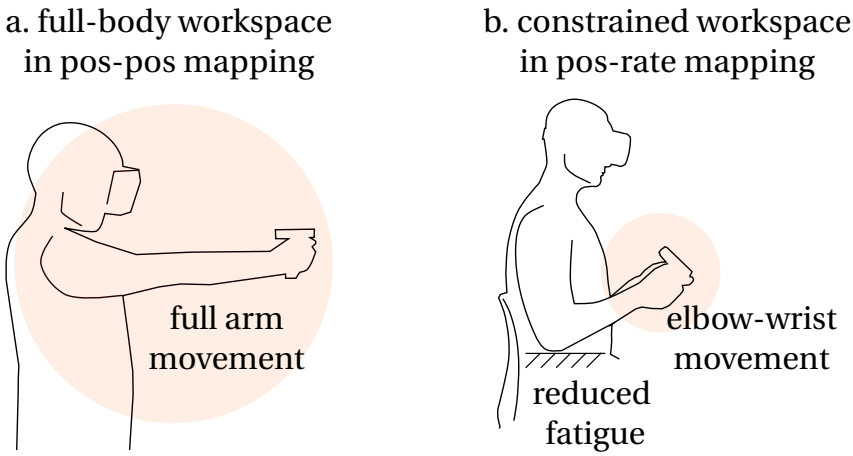


Figure 4.1: Input device workspace comparison in position-position and position-rate mapping modes. Smaller arm movements can be used in rate mode, leading to reduced fatigue: a) the operator is using full arm movements in position-position mapping to control the robot; b) the operator is using elbow-wrist movements, while arm is constrained to control the robot in rate mode.

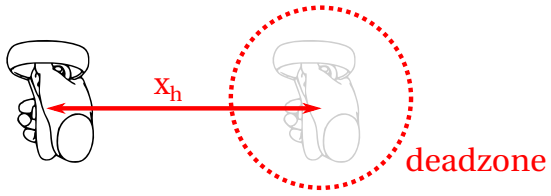
words: "Should virtual world scale affect position-rate mapping gain?" For example, if the operator scales the virtual world down, which is associated with large movements in position-position mapping, would the operator find it more or less intuitive if the position-rate mapping gain would increase similar to position-position mapping, see Fig 4.2.b).

Aim of the study. The goal of the study is to determine which of the rate mode control options leads to lower task execution time and/or workload: a) constant rate mode scaling (independent of the VR-scene scale) and b) variable rate mode scaling that depends on the VR-scene scale.

4.2 The experiment

4.2.1 Experimental task

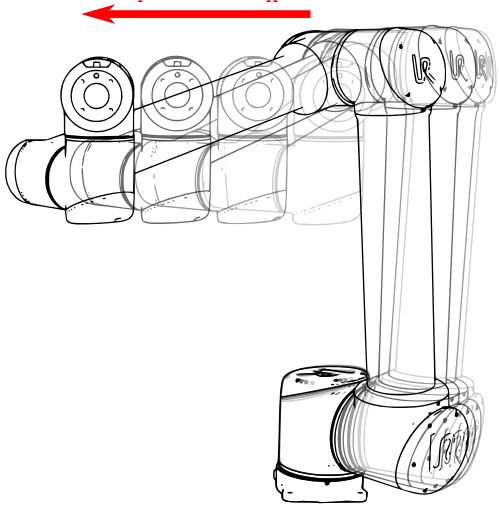
Participants were asked to perform robotic reaching tasks that required controlling the robot's end-effector's speed to reach and stop at a randomly generated reference point (target). The experimental task is shown in Fig. 4.3. Once an end-effector tool touched the target and stopped for longer than 2 seconds a new target was generated. Each



for the same real-world input device displacement, consider:

a) constant rate scaling that, independent of the virtual world scale

$v_r = kDZ(x_h)$



b) variable rate scaling dependant on the virtual world scale, for example: virtual world scaled to a half of real $s^{V \rightarrow R} = 0.5$

$k_{var} = ks^{V \rightarrow R}$

$v_r = k_{var}DZ(x_h)$

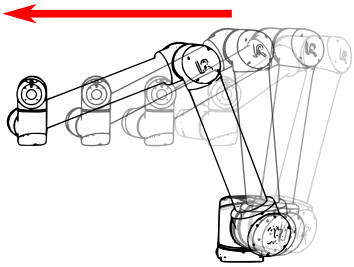


Figure 4.2: Constant and variable rate mode control. For the same displacement of the input device following rate modes can be used: a) constant rate mode, that does not change with the virtual world scale; b) variable rate mode, that changes along with the virtual world scale

target was generated randomly within 20% - 80% of the robot's reachable workspace above the robot's base frame (hypothetical floor). The 20% and 80% limits were imposed such that targets would not spawn too close to the robot to cause potential self-collision scenarios, nor on the edge of the robot's reachable space that would cause the robot to enter into joint-singularity configuration space. Since targets were small and could potentially spawn outside of the operator's field of view, targets would spawn at a large size and quickly shrink - this animation was meant to catch the operator's attention.

In the experiments, participants were seated with their elbows resting against their torso to limit their range of motion, thereby enabling control of the robot using only elbow and wrist movements. The decision to restrict movement was based on the hy-

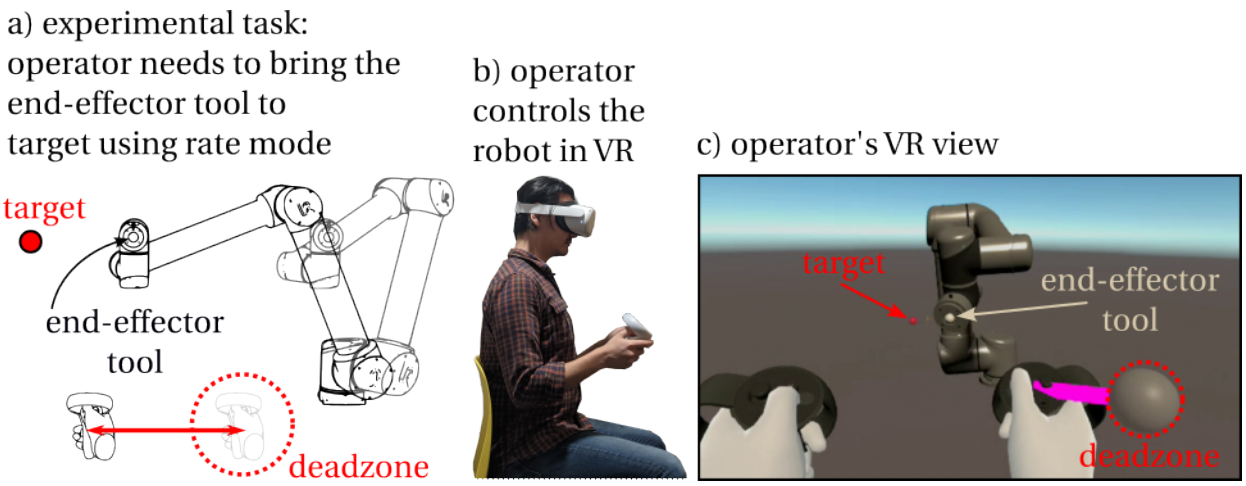


Figure 4.3: The experimental task and experimental setup: a) illustration of the experimental task - the operator needs to bring the end-effector tool to the target using rate mode; b) operator uses Oculus Quest 2 headset and Oculus Touch controllers to teleoperate the robot in VR; c) operator's VR view (similar to illustration in a)).

pothesis that rate mode control would be advantageous in this scenario and that using only the elbows and wrists would result in less fatigue.

The experiment was performed during the COVID-19 lockdown, therefore only five participants (healthy adults; one female; age 25-30) were recruited to perform the experiment. All participants have provided their consent by signing the consent form, as per the ethics approval QMERC20.403. All participants had little prior experience with VR, two participants had prior experience with robot teleoperation.

Each participant did the experiment on two separate days to reduce the effect of fatigue and track learning. On the first day, a participant performed 15 trials with short breaks in-between trials in constant mapping mode and filled in the NASA-TLX questionnaire. On the following day, a participant performed 15 trials in variable mapping mode, followed by the NASA-TLX questionnaire. A single trial contained 10 targets. On average, per day participants spent 40 minutes on the experiment and additional 10 minutes on the questionnaire.

4.2.2 Experimental setup

The experiment was conducted using the VR-based robot teleoperation framework described in chapter 2, with a few adjustments due to COVID-19 lockdown, described here. The experiment was conducted using a simulated robotic setup. A wearable VR

headset with two hand-held controllers (Oculus Quest 2 VR headset, Oculus Touch controllers) were used to teleoperate the virtual robot-manipulator (UR5 robotic arm) simulated with the ROS and Unity 3D Engine (see Fig. 4.3). Participants controlled the robot in constant and variable rate mode control options described in section 2.3. Participants have used gesture-based navigation described in section 2.2.4. Parameters used in the experimental setup were: deadzone size r_{dz}=0.025 m; k=0.1; parameters were defined empirically by the experimenter. The simulation and the VR interface were running at 60Hz.

Relaxed-IK [99] was used as an inverse kinematics solver. The solver was set up such that it would accurately represent a real robot - it took into account the robot's joint limits and their dynamic capabilities (maximum joint angles, velocities and accelerations as well as self-collision configurations) identical to a real robot.

Targets were visualised as animated red spheres. When a target was first generated, it had a 10cm (at $s^{V-R} = 1$) diameter that reduced to 1cm over 1 second. The animation helped participants quickly locate a new target in the scene. The end-effector tool also had a 1cm diameter. A target reaching was considered to be successfully completed if the robot's end-effector remained stationary within at most a 2cm distance from the target's centre for at least 2 seconds (i.e. if there was any overlap between the end-effector tool and a target). Only then the old target was removed and a new target was generated and visualised.

4.2.3 Metrics

In each trial following data was recorded: a target's position and completion timestamps, the robot's end-effector total travel path (as a sum of end-effector displacements over each target completion), the participant's both hands' total travel path (similar to end-effector total travel path), participant's head rotation range in pitch and yaw, the visibility of a target (negative if a target was occluded or outside of the participant's field of view), the virtual world's scale history, the number of virtual world rotations and scale changes.

Target completion times were used as indicators of the overall efficiency of the control mode. The total end-effector travel distance was used as an indicator of motion planning efficiency - longer trajectories are less efficient. Head rotation and visibility of a target were used as indicators of clarity of proposed experimental control interfaces

(i.e. how well participants could see the robot and the target). If participants would use different virtual world scales in experimental modes, the visibility of the target would also likely be different - for example, consider a scenario where the operator zooms-in closely to perform a precision movement to clear a target, then if the target would spawn on the edge or out of operator's field of view the operator would have to rotate their head more. Hence head rotation and target visibility were used to check whether experimental modes made it harder for them to visually locate targets which could contribute to physical and mental loads as well as frustration. Virtual world scale and the number of virtual world scale rotations and scale changes were recorded to analyse whether variable rate mapping would affect participants viewing preferences between modes.

4.3 Results

4.3.1 Learning

Learning curves of task completion time for both modes are shown in Fig. 4.4. Since all participants started with constant mapping mode as their baseline mode on the first day, the learning curve of the variable mapping mode, used on the second day was less steep, as participants adapted to the new control strategy in the first 5 trials. However, in both modes, the learning curve flattened after trial 5 (50 reaching tasks), which indicates that during trials 6-15 there was no significant learning nor deterioration of performance due to fatigue, and the data from trials 6-15 were used for further analysis.

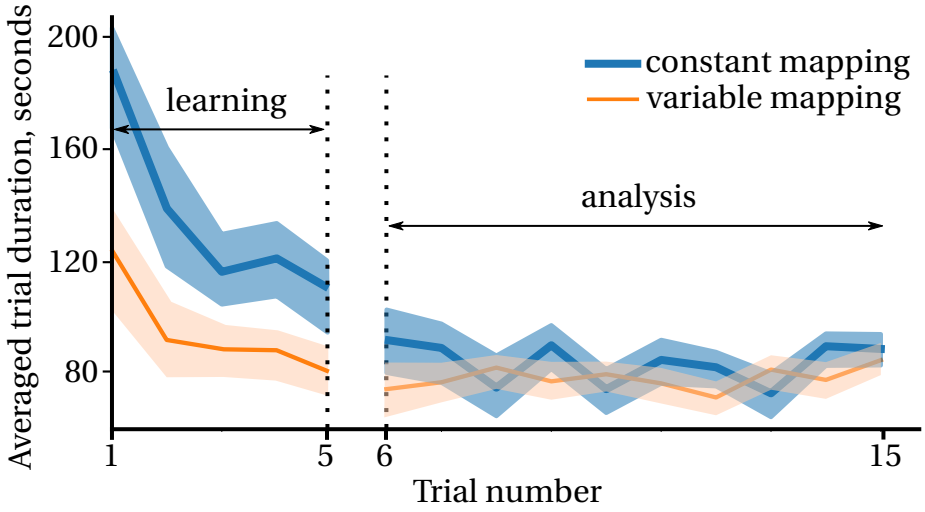


Figure 4.4: Trials' duration (completion time) averaged across all participants. The training curve settles after the first five trials, the latter 10 are used for analysis.

4.3.2 Using rate mode with variable scaling

Fig. 4.5 shows a sample single target completion during teleoperation with variable mapping mode. The sample starts once the old target is cleared and a new target appeared. The participant scaled-down the virtual world from near real world size (s^{V-R}=1) to very small (s^{V-R} >5) and performed a large motion that overshot the target on the x-axis. The participant then scaled the world up to a half real world size (s^{V-R}=2.0) and followed up with a corrective motion on x-axis. Note that both the initial large motion and the smaller corrective motion are generated by two right hand (dominant hand for this participant) displacements that are similar in amplitude but occur at different virtual world scales.

4.3.3 Completion time

The top plot of Fig. 4.6 demonstrates the target completion time for all participants (excluding the 2 seconds that participants had to wait inside the target during the reaching task). It can be seen that the mean is lower in variable mapping with statistical significance for three participants.

The data from each reaching task was split into planning, action and waiting phases: a participant moving the robot is considered as action phase; a participant waiting for

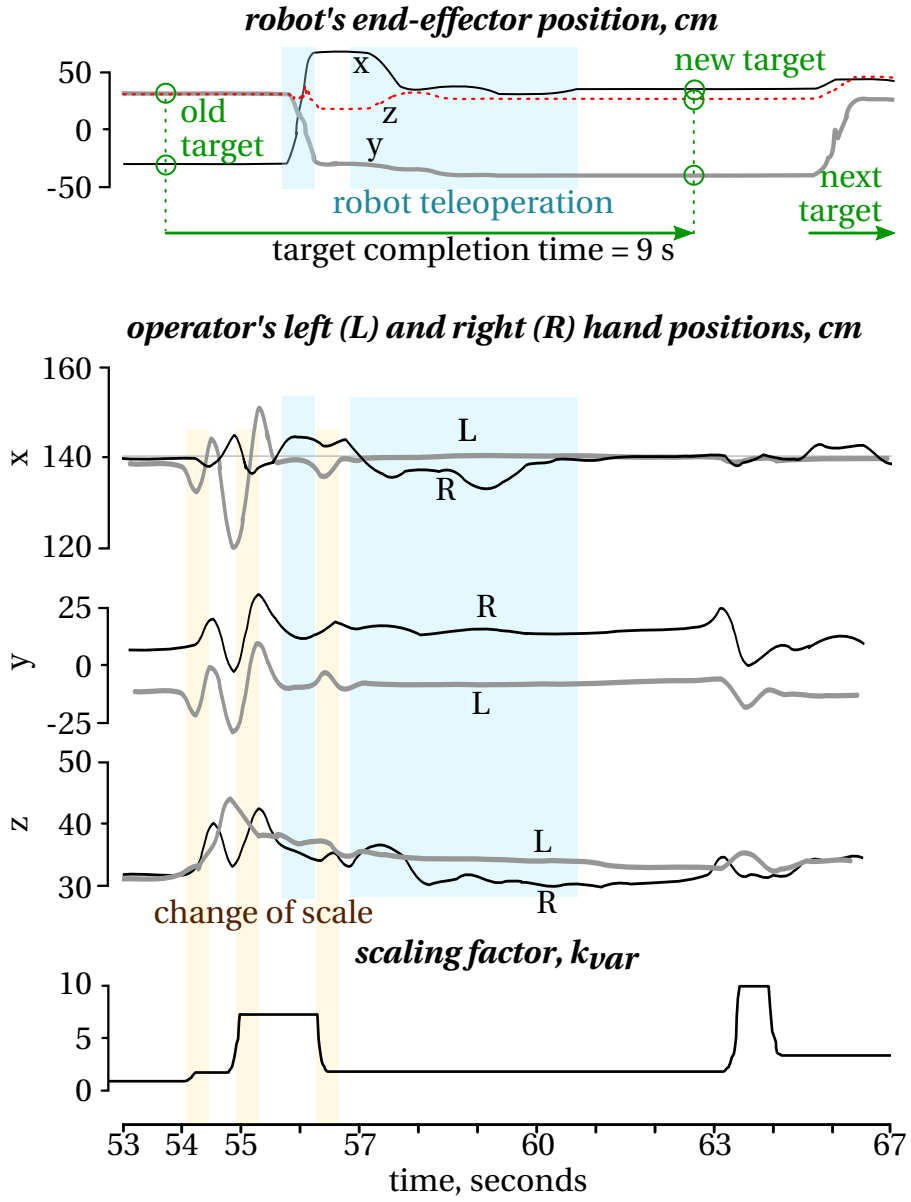


Figure 4.5: Results demonstrating a single target reaching iteration for a typical subject with variable mapping rate mode control. Time histories of the robot's end-effector position (top), the operator's left (L) and right (R) hands (2^{nd}-4^{th} plots) and the scaling factor (bottom plot). The position of the robot and hands are shown in Cartesian space with x-, y- and z- coordinates. Yellow areas highlight the change of scale of the virtual scene. Blue areas highlight robot teleoperation with rate mode control.

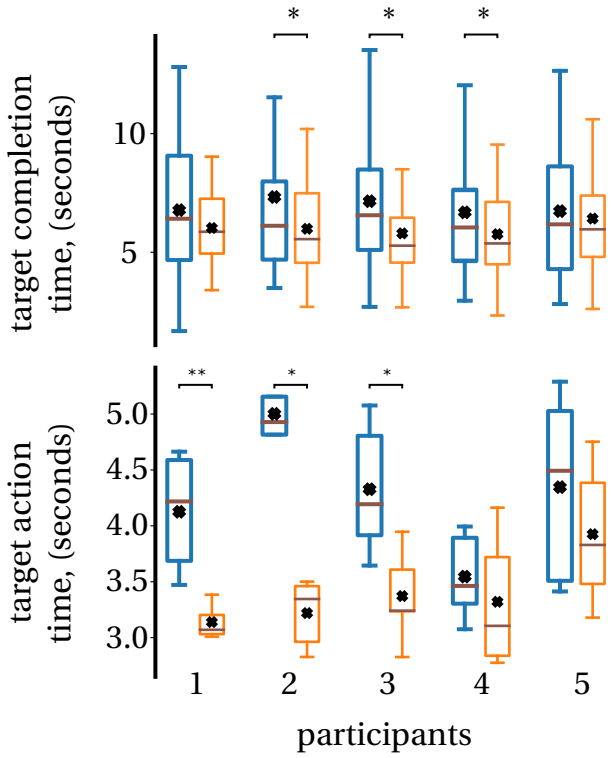


Figure 4.6: Target completion time and action duration per target: box extends from the lower to upper quartile values of the data, with a line at the median and a cross at the mean. The whiskers extend to 1.5 interquartile range from the box. *p<0.05, **p<0.01, T-test

Table 4.1: The summary for performance indicators

	const. mapping M ± SD	var. mapping M ± SD	
Dominant hand total travel distance per target, m	1.28 ± .34	1.13 ± 0.27	
Non-dominant hand total travel distance per target, m	1.0 ± 0.26	1.04 ± 0.27	
End-effector total travel distance per target, m	1.1 ± 0.29	1.12 ± 0.21	
Number of scale changes and rotations per target	2.33 ± 0.48	2.64 ± 0.46	
Number of times target was not visible per target	0.21 ± 0.05	0.23 ± 0.05	
Head pitch range, rad	0.0 ± 0.11	0.0 ± 0.14	
Head yaw range, rad	0.0 ± 0.21	0.0 ± 0.22	
Virtual world scale	0.5 ± 0.1	0.5 ± 0.25	
NASA-TLX	60.5 ± 5.1	68.3 ± 3.2	*

*p $<$ .05, M - mean, SD - standard deviation

a target to be completed with the robot tool inside the target is considered as waiting phase (always 2.0 seconds per target as described above), everything else is considered planning phase. The breakdown of each reaching task into phases showed that participants have spent less time in the action phase in variable mapping mode, with three out of five showing statistical significance (T-test), as shown in the bottom plot of Fig. 4.6.

4.3.4 Head and hands movements

Table 4.1 presents results for the metrics with statistical significance analysed with a T-test. Participants moved hands more to change the virtual world scale than to teleoperate the robot. The dominant hand's total travel distance is a sum of teleoperation motions and scaling motions, meanwhile, the non-dominant hand's total travel distance is only attributed to scaling motions. The difference between the total travel distance between dominant and non-dominant hands is small. Despite some involuntary hand motion due to torso rotation and involuntary hand movement while resting, it is clear that participants moved their hands much more for virtual world scaling/rotation than

for teleoperation. Hence, further studies that focus on reducing physical exertion in VR teleoperation with rate mode need to develop more effective virtual world manipulation and navigation methods.

Total end-effector travel distances per trial are comparable in both modes. Hence, it can be concluded that participants have chosen equally efficient trajectories and movements in both modes. The dominant hand travel distance on average was marginally lower in variable mapping mode but not statistically significant. The non-dominant hand travel distance is the same for both modes. Hence it can be concluded that both modes should cause similar physical fatigue.

The average virtual world scale was similar for both modes with higher variance when scaling was enabled. Participants have arguably made more rotations and scale changes in the variable mapping mode (p=0.09). We believe these results can be explained by participants making use of variable rate mapping to sequence large motions in a scaled-down virtual world and precise motions in a scaled-up world. This results in more virtual world manipulations compared to constant mapping where participants use scale change and rotations only to get a better view of the robot and the target.

In both modes the head pitch and yaw range of rotation were similar. The number of times the target was either outside the field of view or occluded by the robot is similar as well. It can be concluded that target visibility was similar in both modes, and neither mode was more confusing to the participants.

4.3.5 Workload

On average variable mapping mode has scored 13% (p <0.05) higher on NASA Task Load Index compared to constant mapping. It indicates that participants have found the variable mapping mode to be more demanding. In the post-test interviews, participants noted that they had to plan their sequence of actions more thoroughly in the planning phase and move their dominant hand (hand used for teleoperation) more precisely when scaled down in the action phase. This is also evident in the average NASA-TLX breakdown as shown in Fig. 4.7, where the variable mapping mode graph is heavily skewed toward physical and mental demands. By comparison, participants have attributed the workload of the constant mapping mode to the temporal demand and performance (pressure to perform the test accurately and quickly) and frustration.

It could be argued that scheduling the constant mapping mode on the first day and

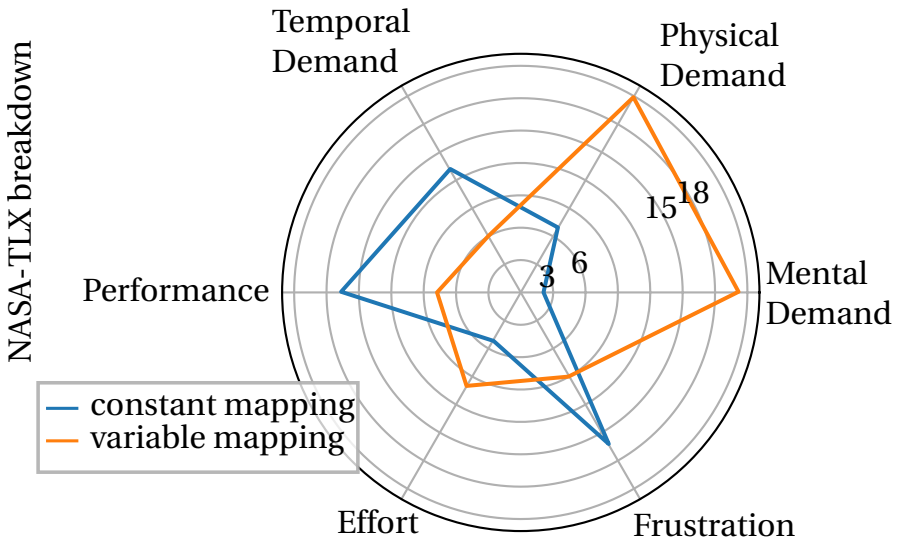


Figure 4.7: NASA-TLX breakdown, averaged across all participants

the variable mapping on the second could have affected the perceived workload and its distribution. On the second day participants arrived with more realistic expectations of their performance and focused more on physical and mental demands as they had to adapt to the new control strategy. It is clear in comparison that the variable mapping is more mentally and physically demanding. Given that total hand motion paths are similar or lower in variable mapping mode it can be concluded that higher physical demand is not caused by the amount of movement but by the accuracy of movement.

4.4 Discussion

It was expected that participants would find the variable mapping more intuitive as the position-speed (rate) mapping gain was adjusted along with the scale of the robot's visualisation in VR, as opposed to control and visualisation being independent in the constant mapping mode. The experimental results partially met the expectations. Participants managed to perform reaching tasks with teleoperated robot faster in rate mode with variable mapping, as participants were able to better control the robot in large and small motions, resulting in shorter task completion time. Variable mapping was less beneficial for reducing the workload as indicated by the increased overall NASA-TLX score. Participants had to plan their movements more thoroughly and move more pre-

cisely (indicated by higher mental and physical demand in the NASA-TLX breakdown). The study showed no increase in participants' experimental task completion time by the end of 40 minutes of the experiment. However, it is unclear if participants' performance could drop in longer teleoperation sessions due to fatigue. No difference in motion planning efficiency during teleoperation was observed. The total amount of participant hands and head movement were not statistically different for variable and constant scaling modes. Participants were able to view the robot's workspace equally well in both modes.

The study was limited by a small participant sample, however, it was designed to include a large number of task repetitions to compensate for it. Further studies can benefit from an expanded participant sample pool. One also needs to consider the effect of non-randomised study order on results - participants performed the experimental task in constant scaling mode on day 1, and variable scaling mode on day 2, such that constant mapping mode was established as the baseline. As a result, the learning curve of variable scaling mode was less steep compared to constant mapping mode as shown in Fig. 4.4. However, it should be noted that experiment analysis only used data from trials 6 to 15, where the learning curves have settled for both experimental modes, minimising the effects of transferred learning from mode to mode. But it can be argued that there was an unforeseen effect of trial orders on participants' NASA-TLX scores as participants may have had different expectations of their performance on day 2 compared to day 1. Hence in future studies, it is strongly advised to randomise the order of trials.

4.5 Conclusion

This chapter presented a study on the effect of virtual-to-real world dynamic scaling on an operator's ability to teleoperate a robot in rate mode. Both rate control and dynamic virtual world scaling can be used as a solution to size mismatch between the input device's and robot's workspace. The study examined whether these two methods would complement or impede each other. The study has demonstrated that participants found it more intuitive for rate mode gain to adjust along the virtual world scale rather than for visualisation and control to be independent. That said, the results need to be taken with a grain of salt given that the participant pool size was rather limited as the experiment was conducted during the COVID-19 lockdown. However, the study provides a convenient jumping-off point for further studies with mobile robots.

5 Human-operator's visual preferences in VR-based robot teleoperation

Contents

Chapter summary: This chapter reports on a study that explores the learning process of operators in using VR-based teleoperation, tracking their progress from novice to expert levels. Additionally, the study examines how the visual preferences of operators evolve over time, and how prior experiences of participants affect their learning process and visual preferences. The focus of the study is on the virtual-to-real remote reconstruction scale and visual attention, specifically in a pick-and-place robotic task. The study found that expert teleoperators, as a group, tended to use a smaller virtual world scale compared to novices, although this behaviour was not universal among individual teleoperators. Prior video gaming experience was found to affect virtual world scale, with experienced gamers using smaller virtual world scales and making fewer scale changes. Regarding visual attention, the study found that operators' gaze distribution was similar to that of a real world pick and place task, with most attention focused on the manipulation object and pick and place locations. However, unlike real world manipulation tasks, operators tended to focus more on their manipulation hand (dominant hand) despite its differing functionality in the robotic task. The study also demonstrated that prior video gaming experience affected the speed and accuracy of operators' task performance and gaze distribution. Finally, the study identified the most common gaze paths and patterns for each teleoperation phase, providing insights that could inform the design of visual aids for operator training.

5.1 Chapter introduction

One of the advantages of VR-based robot teleoperation over conventional display and keyboard setups is its immersive visualisation of the remote environment. However, the most common method for visualising the remote environment - point clouds - can suffer from distortions and occlusions, as shown in section 1.5.4. In Chapter 3, we present methods that alleviate these problems, and one can also note other important works in this field [3, 71, 62]. Yet, another equally important question is what operators actually look at during teleoperation and how we can ensure that we focus on the most important aspects. Therefore, this chapter examines operators' visual preferences during VR-based robot teleoperation.

First, we aim to learn what operators look at during different stages of teleoperation, so we can understand which parts of the visualisation require further attention. To the

best of author's knowledge, no other studies have investigated visual preferences during VR-based robot teleoperation. In related works, Berton et al. [176] demonstrated that gaze behaviour in VR is similar to the real world during human collision avoidance with a walker. Hence it can be hypothesised that the gaze distribution during teleoperated pick-and-place from VR may be similar to that in the real world: Lavoie et al. [174] presented visual attention in real world pick-and-place task. If proven true, the knowledge from real world tasks can be transferred to VR-based robot teleoperation to further improve VR-interfaces.

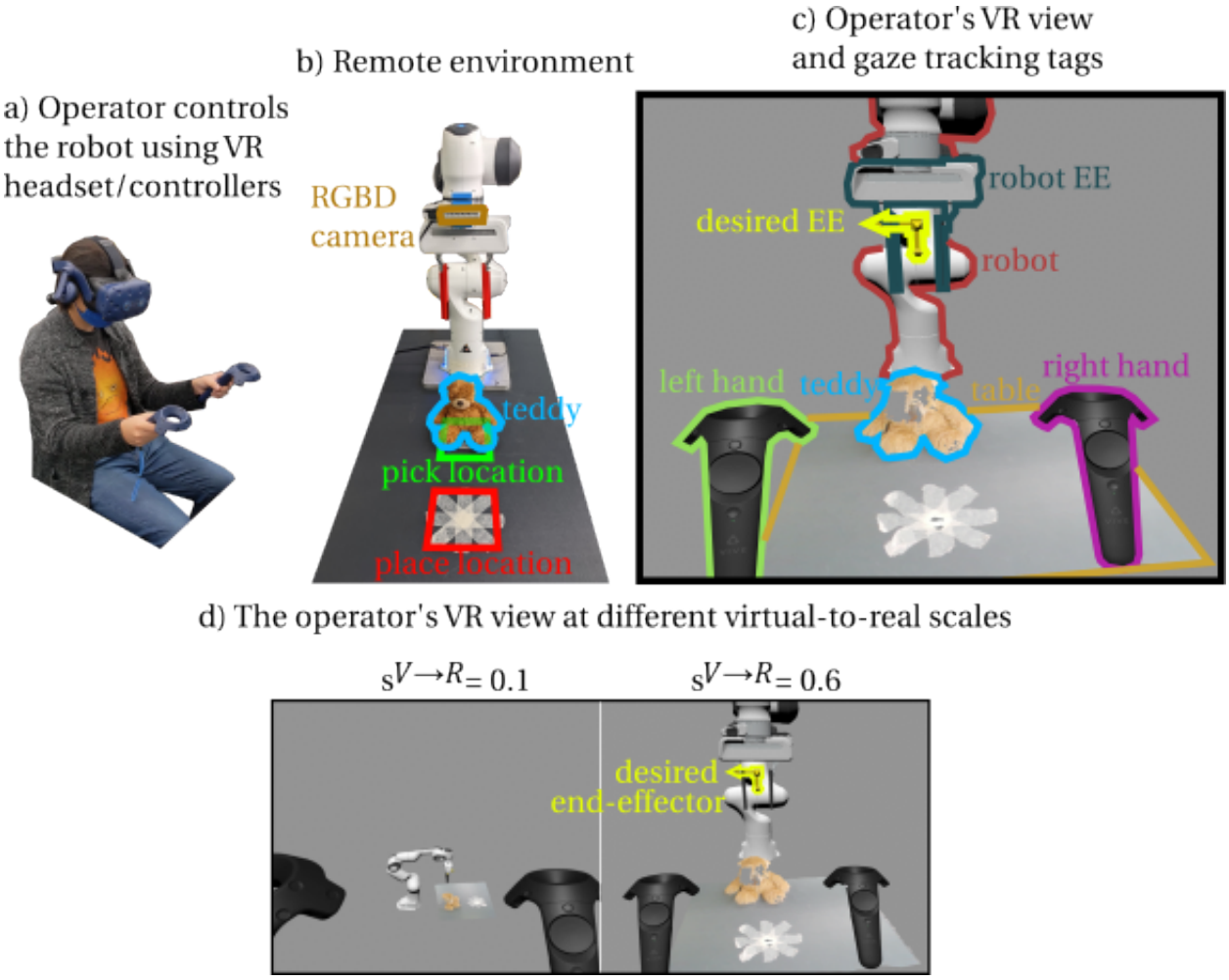


Figure 5.1: Experimental setup: a) human operator with VR headset and controllers, b) remote environment, c) operator's VR view and gaze tracking tags, d) operator's VR view at different virtual-to-real scales

Second, we aim to understand which visual preferences are more beneficial for robot teleoperation. Visualisation control within the VR environment is also specific to each human-operator depending on how they decide to navigate and it depends on their prior experience [236]. Teleoperation performance is positively affected by the video game experience an operator has. For example, it was shown that video gaming experience is associated with a higher baseline performance in laparoscopic simulator trainer skills [237]. Understanding the differences between good and poor teleoperators can be used to improve teleoperator training procedures, and the key to understanding said differences may lie in their visual attention distributions.

Another important aspect to consider is how operators view the remote environment. We have previously introduced gesture-based navigation as the primary method for operators to navigate the virtual world. One of the key features of this navigation method is the ability to manipulate the virtual world scale. Therefore, in addition to the visual attention of operators, we should also consider the visual scale during teleoperation. Is there an optimal virtual world teleoperation scale that can improve teleoperation efficiency? However, to the best of our knowledge, studies on a virtual world scale during teleoperation are extremely rare, with the exception of our contemporary Thomason et al. [66] who proposed a similar navigation system, and the study presented in Chapter 4. Thus, this study also investigates scale usage during teleoperation to contribute to the knowledge base on this important aspect of VR-based robot teleoperation.

5.2 Experiment design

5.2.1 Experimental task and protocol

The experimental task was a single pick-and-place task performed in supervised teleoperation mode. The robot's remote environment contained a manipulation object - a teddy bear placed on a remote environment background - a table. Participants were asked to move the teddy from the pick location to the place location, see Fig. 5.1. In each trial, the robot and the teddy were placed at the same initial position. In VR the operator would also always start from the same initial position (subject to participant's height) facing the back of the robot. Participants teleoperated the robot whilst sitting.

15 participants were recruited (8 male, 7 female) from Queen Mary of London University graduate and post-graduate students. We separated participants into three groups

based on their prior gaming experience: **N** - no experience (n_N = 6 participants), **M** - medium experience (n_M = 4) and **H** - high experience (n_H = 5) based on a self-reported questionnaire (included in supplementary materials). Four participants reported normal vision, another four performed the experiment whilst wearing contact lenses, another seven reported that they wear glasses but can perform daily routine activities without them. Participants performed the experiment on days 1, 2, 7 in order to track short-term and long-term learning. It was assumed that on day 1 all participants are novice operators and by day 7 all participants will be expert operators.

Prior to starting the experiment, each participant filled in a consent form in accordance to the ethics approval QMERC20.403 issued by Queen Mary University of London ethics board. The experimenter demonstrated the experimental task and explained the VR interface controls. Participants then performed the G2OM's (Gaze-to-object-mapping) gaze tracking calibration and trained for 5 minutes. The gaze tracking calibration was performed using the procedure described in section 2.4.2, once participant performed the calibration the experimenter verified the quality of gaze tracking by asking the participant to use their gaze to light up all the dots on the verification screen (see Fig.2.11). During training the experimenter has verified that participant can navigate in the virtual space, control the robot, and grasp/release an object. Then participants performed the experimental task.

On each experiment day we set 40 minute limit (excluding breaks midway) to avoid effects of fatigue; within this limit each participant had performed on up to 10 trials. During the experiment individual trials would be considered failed by the experimenter if they did not meet the expected baseline sequence of actions, discussed in more detail in section 5.3.1. In those cases, participants performed extra experiments to compensate, up to the aforementioned time limit.

Following data was recorded at 40Hz on every VR visual update frame: gaze vector, gaze ray (origin and direction), gazed object's id, operator's head and hands poses in VR and real space, VR world visualisation scale, robot's state. Additionally timestamps of operator's requests to move the robot, open and close were recorded in order to determine starts and ends of teleoperation phases. Finally, trial durations were recorded as well.

5.2.2 Experimental setup

The experimental setup was based on VR-based robot teleoperation framework introduced in chapter 2. It consisted of a Franka Emika's Panda robot with an end-effector mounted Intel id435 RBGD camera, HTC Vive Pro Eye VR headset and controllers that performed gaze tracking using the Tobii Gaze-to-object-mapping (G2OM) machine learning model in Unity. The details of the gaze tracking is described in section 2.4.

OctoMap was used as a rough meshing solution for point cloud, such that gaze tracking can be performed on the point cloud. OctoMap was not visualised in VR interface, and only served as invisible point cloud bounds. OctoMap was tuned to "forget" nodes if not observed for longer than one second. This reduced the possibility of OctoMap nodes that are not currently observed by the camera, i.e. nodes that do not have live point cloud contained within them catching participant's gaze. OctoMap nodes were separated into "table" and "teddy" tags based on nodes height along the z-axis.

The robot was controlled in supervised teleoperation mode as described in section 2.3.2. The operator would use task specific grasp-on-pose control, where operator sets the desired end-effector (intractable axis mesh in Fig. 5.1) position at grasp pose. The robot would then immediately plan and perform a corresponding grasp. Note that trajectory preview was excluded from this teleoperation mode to reduce task execution time.

5.3 Results

5.3.1 Dataset distribution

Prior to analysis the recorded data was split into valid trials, outliers and failed trials. The sequence and numbers of failed and outlier trails is presented in Fig. 5.2. A trial was considered valid if it followed a baseline of actions: participant navigates virtual world and positions themselves such that the robot and the remote environment are clearly visible and robot can be teleoperated, a valid grasp pose is set and performed successfully on the first try, place pose is set and executed on the first try.

Failed trials were flagged during the experiment by the experimenter if one of the following occurred: a) object slipped from the gripper during manipulation if participant made a shallow grasp, b) the robot's motion planning failed if participant set target end-effector pose that would violate robot's joint and/or collision limits, c) rosbag

(ROS-based message recording method) recording failed due to rosbridge TCP/IP error. In first two scenarios action and gaze distributions would vary too much from the baseline scenario, hence they were considered failed trials. In the last scenario, either parts of data were missing, if rosbridge dropped and then re-established communication or rosbag and messages' internal timestamps mismatched dramatically. Post-experiment the latter cases were salvaged as all recorded rosmessages had own message timestamps synced by the VR-interface; this however resulted in slight imbalance in dataset - day 2 had 14 "unfailed" trials.

Post-experiment trials were considered outliers if a participant had to move the robot more than 2 times to successfully pick and place the object, for example if participant set a pick pose, changed their mind and set a new one, etc. Similar to failed trials these trials would present gaze and action distribution considerably different from the baseline; hence they were excluded from further analysis.

Outlier and failed trials occurred more often at the starts of experiment days. Less valid trials were recorded in day 1 as there was a larger number of failed and outlier trials and participants performed the task slower, performing less trials before running into the daily time limit. It should be noted that on day 2 164 (155 valid + 9 outliers) trials were recorded instead of expected 150 as some valid failed trials were salvaged as mentioned above. Overall once can observe steady decrease of failed and outlier trials on subsequent days.

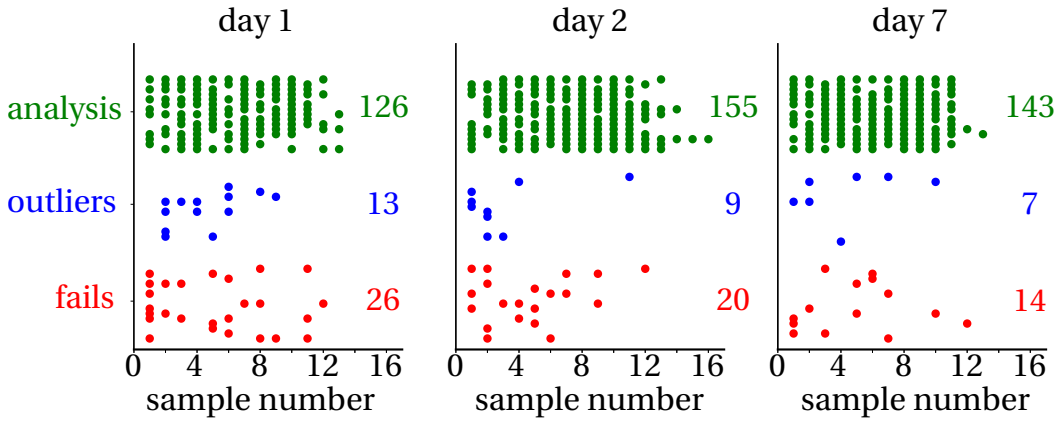


Figure 5.2: Distribution of successful, failed and outlier trials. Participants have completed the least number of successful trials on day 1 before running into the time limit.

5.3.2 Task execution time learning curves

Each trial can be generally broken down into 3 phases: exploration, pick and place. Exploration phase started from the first gesture-based navigation action of the participant until participant's first interaction with the desired end-effector (Fig. 5.1 - yellow axis mesh that is manipulated by the operator to set desired end-effector pose). In the exploration phase the participant navigated the VR space and placed themselves in front of the robot within arm reaching distance. The pick phase started from the participant's first interaction with the desired end-effector until the teddy was grasped. In pick phase the participant would set the desired grasp position and request robot to move and grasp. The place phase starts from grasp execution until the target object is released. In place phase the participant would set the desired end-effector at target place position, move and release the object.

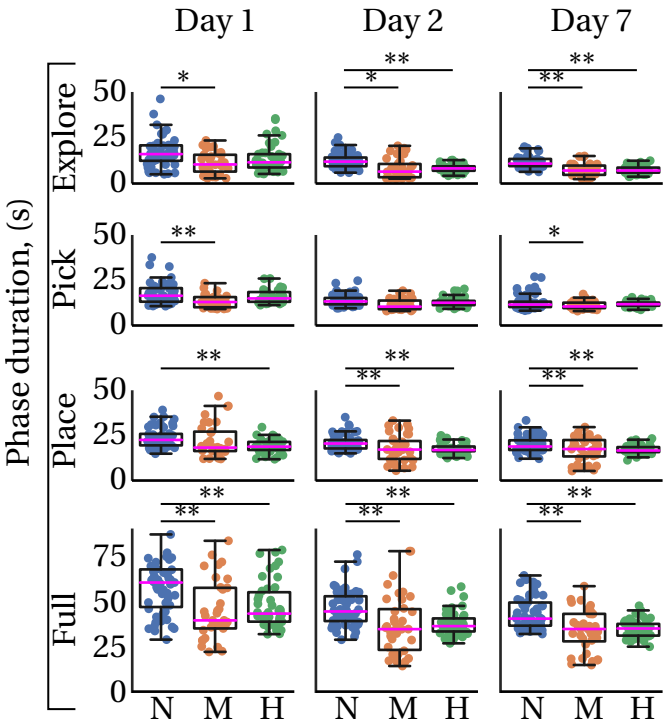


Figure 5.3: Task execution time comparison split by experiment days, teleoperation phases and participants' videogame experience. Boxes extend from the lower to upper quartile values of corresponding data, with a line at the median, whiskers extend to 1.5 interquartile range;*p < 0.05, **p<0.01 Tukey HSD.

Fig. 5.3 presents the comparison of averaged task execution times of participants spilt by experiment day, teleoperation phase and participants' videogaming experience. Participants in **M** and **H** groups often showed, faster (one-way ANOVA, with post-hoc Tukey HSD) task execution times than participants in **N**-group. No statistical difference between **M** and **H** groups' task execution time was observed.

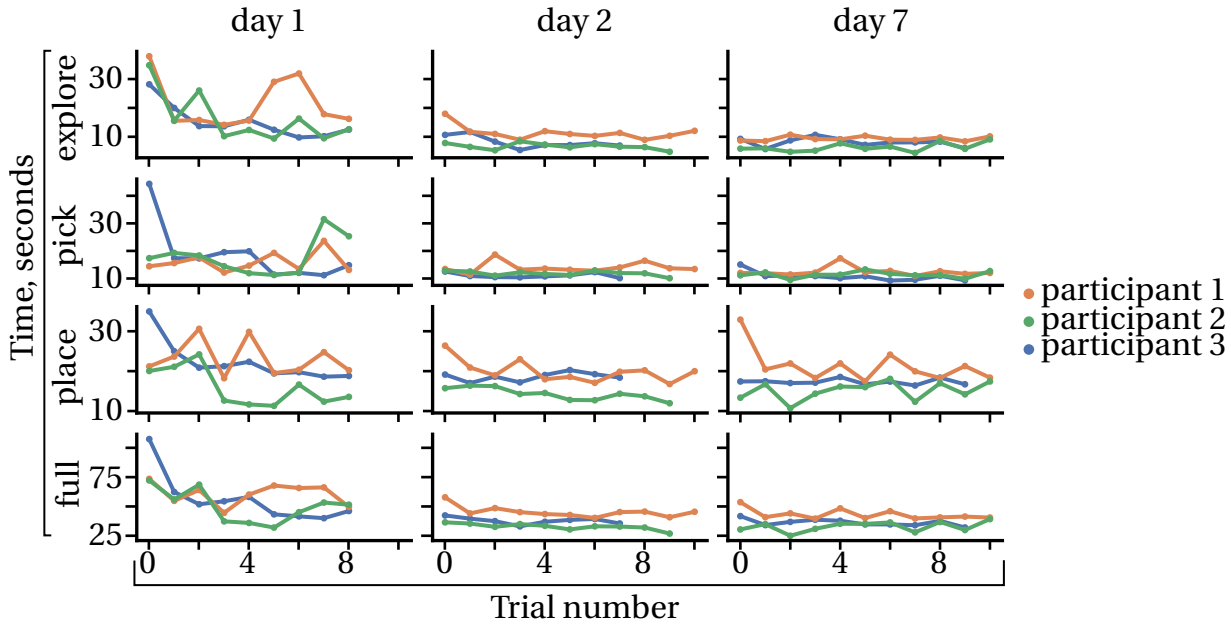


Figure 5.4: Task execution time learning curves of 3 random participants.

Participants' task execution time decreased monotonously for most participants - randomly sampled participants' learning curves are given in Fig. 5.4. For every participant a one-way ANOVA test with post-hoc Tukey HSD test (target p-value < 0.05) was performed on daily task execution times, to check whether participants were improving their task execution times. The Fig. 5.5 presents the distribution time improvements between days, split by participant groups and teleoperation phases. The Table 5.1 shows details time improvement for all participants.

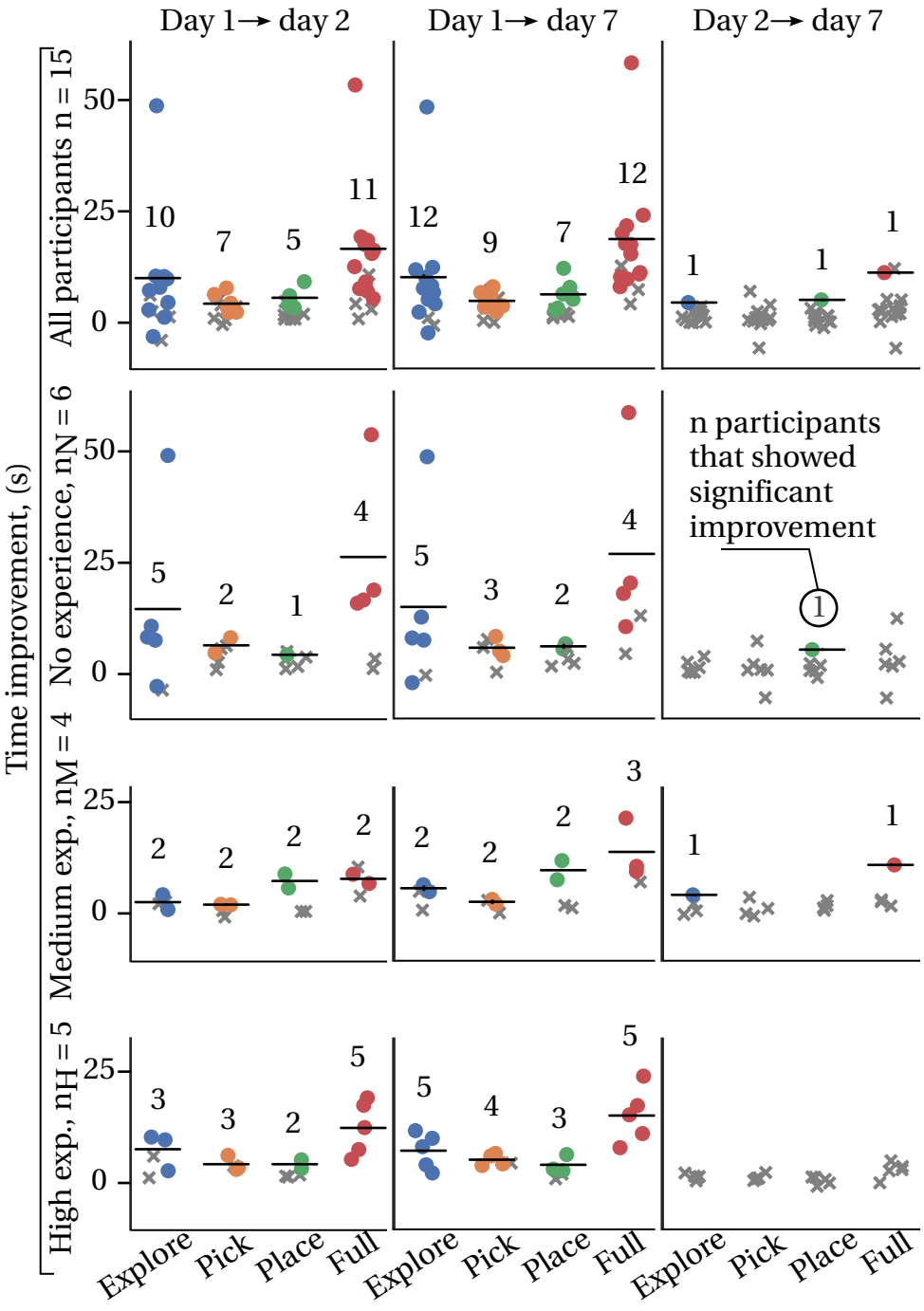


Figure 5.5: Number of participants that showed statistically significant time improvement of task execution time ($p < 0.05$ one-sided ANOVA and post-hoc Tukey HSD) between days, split by participants' prior videogame experience, and teleoperation phase: Coloured dots represent participant that showed statistically significant time improvement, grey crosses represent participants that showed no statistical significance. Numbers over distribution are the counts of participants that showed statistical significance. Black horizontal lines show mean values of statistically significant participants.

Table 5.1: Number of participants N that showed statistically significant time improvement between experiment days and corresponding time improvement means μ (seconds) and standard deviations σ (seconds)

	Day1-Day2			Day1-Day7			Day2-Day7		
	N	μ	σ	N	μ	σ	N	μ	σ
Explore	10	9.8	13.6	12	10.0	12.2	1	4.3	0
Pick	7	4.0	1.9	9	4.7	1.7	0	NA	NA
Place	5	5.4	2.1	7	6.1	2.9	1	4.9	0
Full	11	16.4	12.6	12	18.6	12.9	1	11.0	0

The full task execution time has decreased on average by 19 seconds from day 1 to day 7 for 12/15 (12 out of 15) participants and 16 seconds from day 1 to day 2 for 11/15 participants. Only one participant has showed statistically shorter execution time between day 2 and day 7.

On average participants in **N**-group decreased their task execution time more than medium experience and high experience participants. They also improved most in exploration phase of teleoperation – 15 seconds by 5/6 participants between days 1 and 7. **M** participants improved most in place phase - 2/4 participants by 10 seconds between days 1 and 7. The **H**-group improved the most in exploration phase – 5/5 participants by 8 seconds between day 1 and day 7. Interestingly they were also the only group that has consistently decreased the full teleoperation time - 5/5 participants have showed statistically significant time improvement.

The presence of statistically significant monotonous learning slopes within individual days and in combination of days per participant was investigated using Mann-Kendall test with target p-value < 0.05. The Fig. 5.6 presents the distribution of statistically significant time improvement slopes, split by participant groups and teleoperation phases. The Table 5.2 presents time improvement slopes details for all participants.

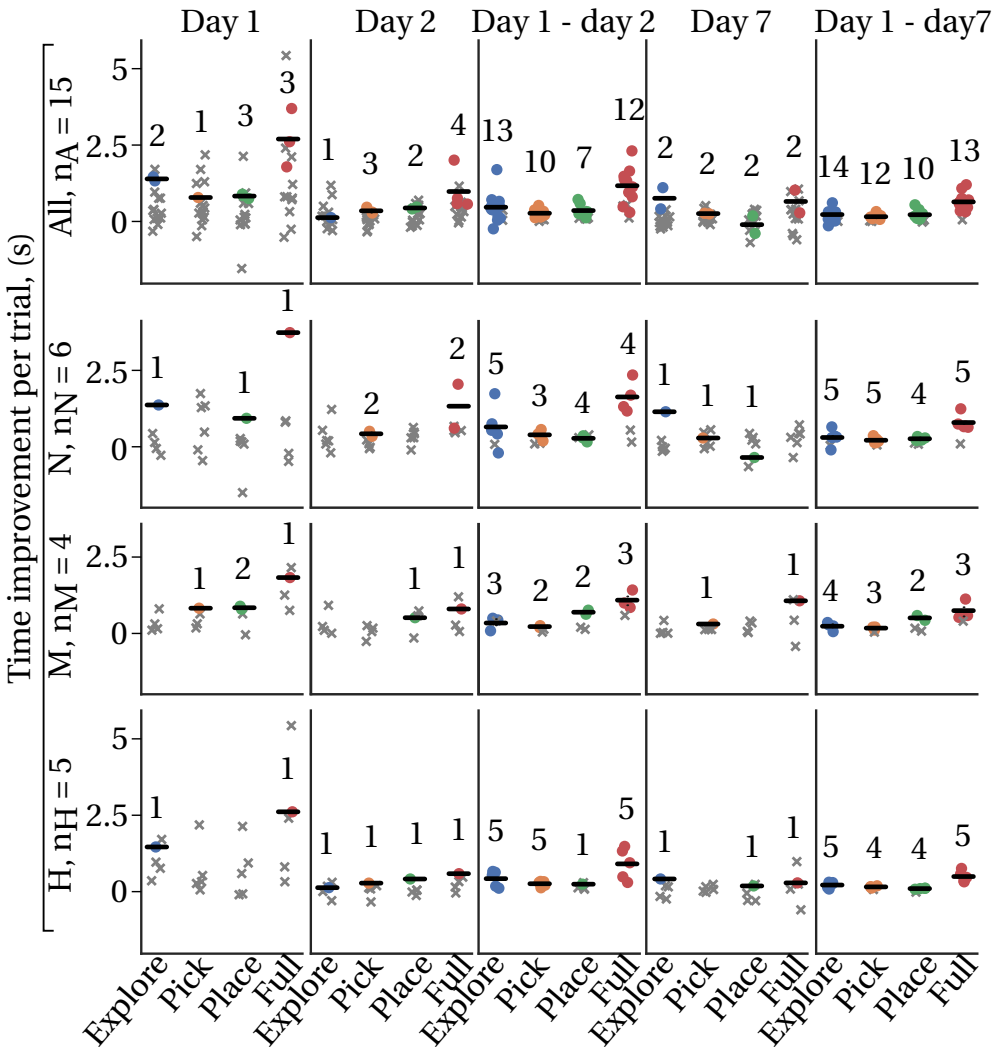


Figure 5.6: Number of participants that showed statistically significant slopes of task execution time ($p < 0.05$ Mann-Kendall) between days, split by participants' prior videogame experience, and teleoperation phase. Coloured dots represent participant that showed statistically significant time improvement, grey crosses represent participants that showed no statistical significance. Numbers over distribution are the counts of participants that showed statistical significance. Black horizontal lines show mean values of statistically significant participants.

Only a few participants showed monotonously learning slopes within individual experiment days – at most 4/15 in day 2 full trial. However, combining successive days, (for example day 1 + day 2) a similar behaviour to Tukey HSD test was observed, for example: 12/15 of participants showed presence of learning curve slope in combined day

Table 5.2: Number of participants N that showed statistically significant time improvement slopes within experiment days and combination of days and corresponding time improvement slope means μ (seconds/trial) and standard deviations σ (seconds/trial)

	Day1			Day2			Day1-Day2			Day7			Day1-Day7		
	N	μ	σ	N	μ	σ	N	μ	σ	N	μ	σ	N	μ	σ
Explore	2	1.4	.06	1	.14	0	13	.47	.43	2	.76	.34	14	.24	.16
Pick	1	.8	0	3	.36	.08	10	.28	.12	2	.27	.01	12	.17	.06
Place	3	.8	.83	2	.45	.03	7	.37	.2	2	-.09	.028	10	.23	.14
Full	3	2.7	2.7	4	0.98	.6	12	1.17	.51	2	.66	.037	13	.65	.24

1 + day 2 and day 1 to day 7.

5.3.3 Virtual-to-real scale effect on execution time

Fig. 5.7 demonstrates a virtual world scale from a randomly picked trial. In this instance, the participant changed their scale twice - grey areas represent scale changes. In nearly all trials, participants were adjusting the scale mainly in the exploration phase, a few participants also adjusted the scale in later phases. In further analysis, the average scale (dotted lines) of full trials and teleoperation phases is used. The scale distribution across all participants and teleoperation phases was found to be normal using the Shapiro-Wilk test.

Fig. 5.8 shows how scales (bars) averaged across all participants along with task execution time (line) as trials went on. The average virtual world scale decreased with trial duration following the same trend, with exception of a spike in trial 14. A closer examination of individual participants trials 14 showed that the spike was caused by higher than average task execution time of three participants in **N**-group.

The further breakdown by participants' video gaming experience and experiment days is shown in Fig. 5.9. All participant groups had a large reduction in scale from day 1 to day 2, for example participants in **N**-group have reduced their average scales from 1.54 on day 1 to 0.97 on day 2. Participants showed little scale reduction from day 2 to day 7. Corresponding trial durations decreased similarly across experiment days as already established in previous section. The **N**-group used statistically significantly larger - scales on all experiment days and only reached the behaviour of participants with video gaming experience by day 7 as demonstrated by one-way ANOVA test with

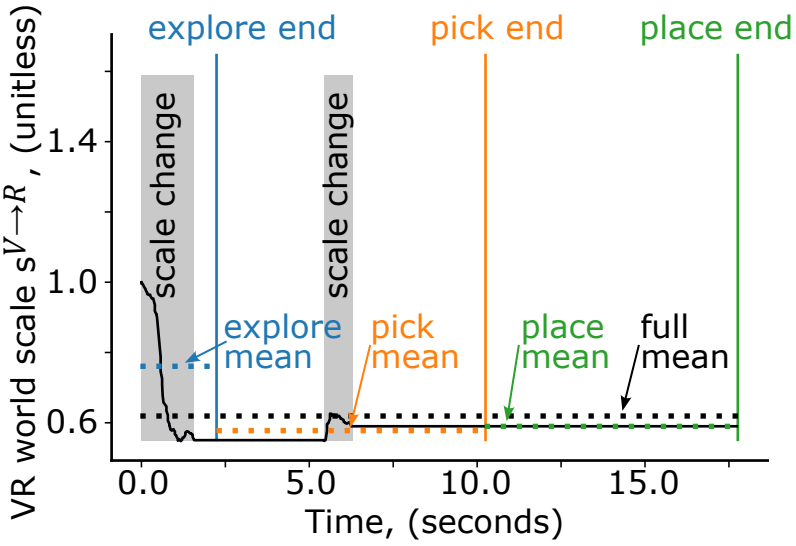


Figure 5.7: Virtual world scale in a sample trial; grey areas show the scale change, solid coloured lines indicate teleoperation phase ends, dotted coloured lines show mean scale per phase.

post-hoc Tukey HSD shown in Table 5.3. **M** and **H** participant groups showed very similar behaviours to each other.

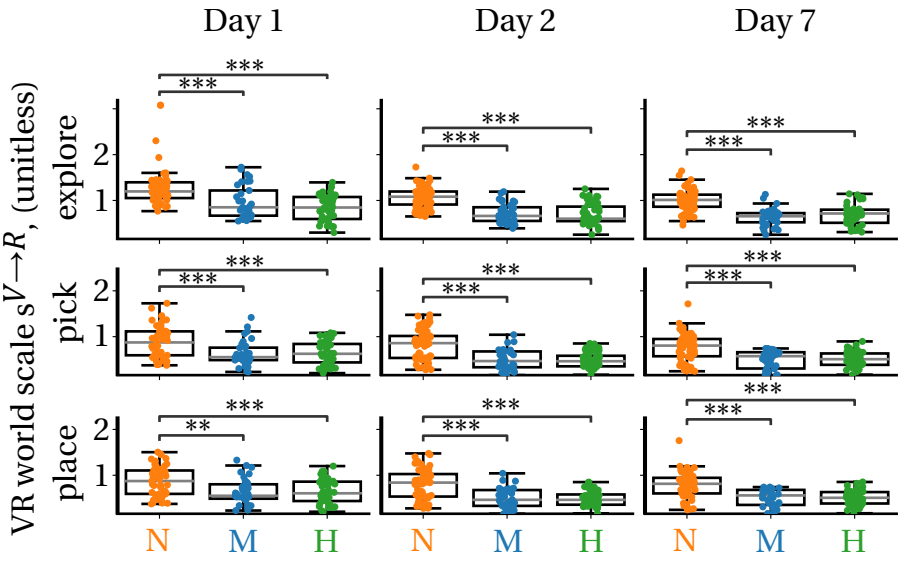


Figure 5.10: Virtual world scale breakdown by teleoperation phase. **p<0.01, ***p<0.001 Tukey HSD

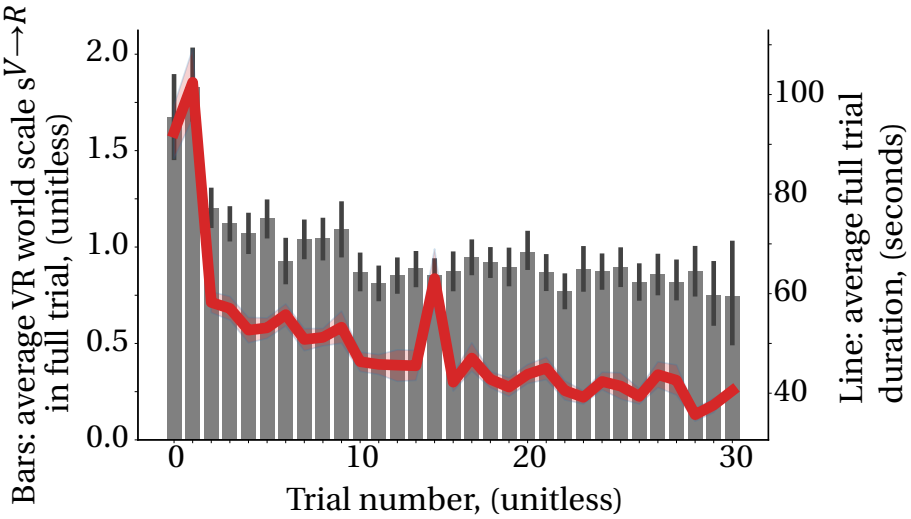


Figure 5.8: History of full trial virtual world scale averaged across all participants. Bars show the average scale on a specific trial, the red line shows the averaged full trial duration.

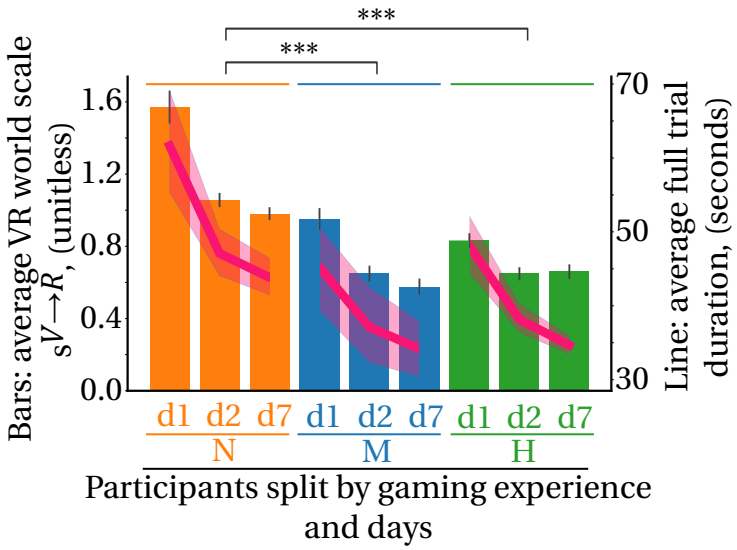


Figure 5.9: Virtual world scale (bars) and duration per trial (line) averaged within participant groups and days. **p<0.01, ***p<0.001 Tukey HSD

Table 5.3: Tukey HSD of scales used by participants on separate experiment days

Day 1					
Group	mean scale	Group	mean scale	mean difference	p-value
N	1.54	**M**	0.95	-0.59	0.001
N	1.54	**H**	0.83	-0.71	0.001
M	0.95	**H**	0.83	-0.12	0.86
Day 2					
Group	mean scale	Group	mean scale	mean difference	p-value
N	1.05	**M**	0.64	-0.41	0.001
N	1.05	**H**	0.64	-0.41	0.001
M	0.64	**H**	0.64	0.004	0.9
Day 7					
Group	mean scale	Group	mean scale	mean difference	p-value
N	0.97	**M**	0.57	-0.4	0.001
N	0.97	**H**	0.65	-0.32	0.001
M	0.57	**H**	0.65	0.08	0.88

Fig. 5.10 shows the further breakdown by teleoperation phases. Again, **M** and high **H** groups performed remarkably similarly in each phase and showed no statistical difference between their respective virtual world scales, as opposed to the **N**-group. Among teleoperation phases, the exploration has the highest average scale. For example in **N**-group participants used average scale of 1.65 in exploration phase and 0.93 and 0.92 in pick and place respectively.

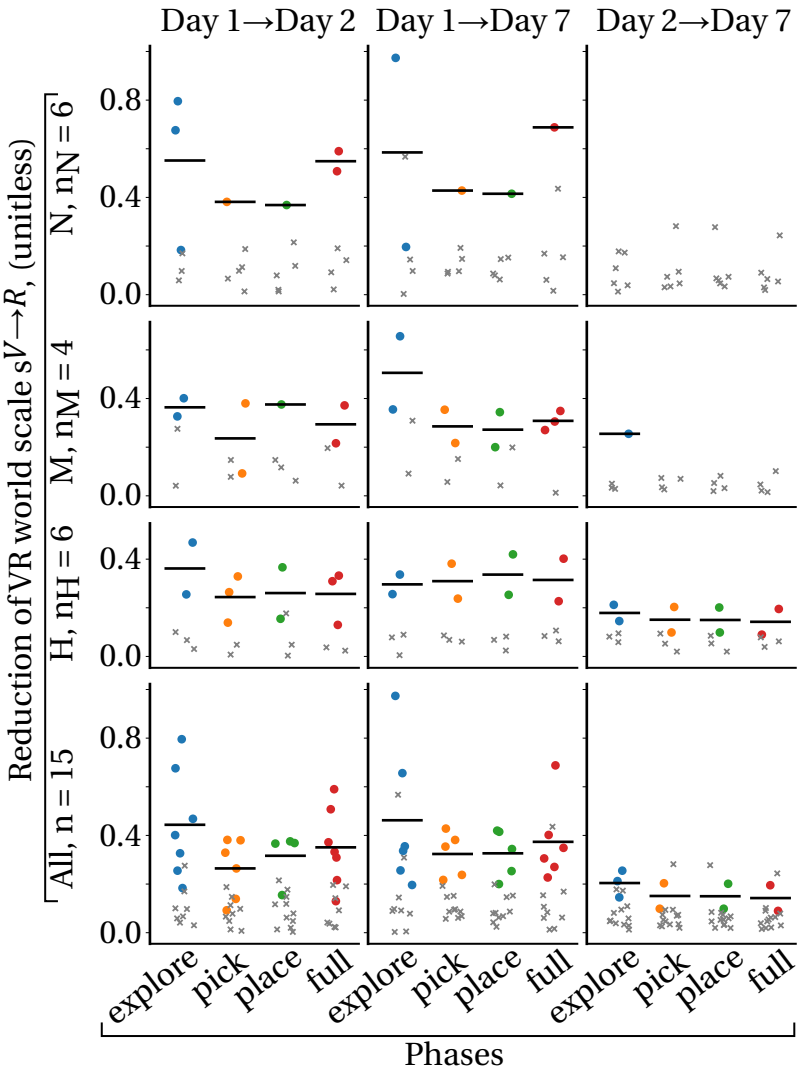


Figure 5.11: Per-participant reduction of virtual world scale between days split by teleoperation phases and participant's gaming experience groups. Coloured dots represent participants that showed a statistically significant reduction in scale change between experiment days, grey crosses are statistically insignificant. Solid black lines represent means of scale reduction among participants that showed statistically significant reduction

Although participants reduce their scales as groups, on an individual level this behaviour is not as common as one might expect. For each participant, a one-way ANOVA test with a post-hoc Tukey test (target p-value < 0.05) was conducted on the average scale differences to determine whether participants have reduced their virtual world scales individually. The scale reduction between days on the per-participant basis is

shown in Fig. 5.11. From day 1 to day 2 only 7 out of 15 participants showed a statistically significant reduction in the virtual world scale and 6 out of 15 from day 1 to day 7. A similar distribution was exhibited within video gaming experience groups as well. **M**-group showed the most consistent behaviour - 3 out of 4 decreased the scale from day 1 to day 7 in full trials. Among phases, the exploration had the most consistent scale reductions.

Table 5.4 presents the summary of scale change behaviours. Overall participants used fewer scale changes as they got better at teleoperating the robot. Participants with no video game experience were the most active in changing their virtual world scale and reached the performance of day 1 gamers by day 7. Remarkably the results show that participants **M**-medium experience adjusted their scales less than participants **H**-high experience, although slower.

Group	Day	Average number of scale changes per participant	Average single scale change	Min scale	Max scale	Average scale change duration, (seconds)
N	1	96	0.24	1.54	1.82	0.95
	2	83	0.18	1.0	1.21	0.85
	7	73	0.18	0.94	1.15	0.82
M	1	50	0.2	0.93	1.18	1.07
	2	41	0.21	0.64	0.9	1.12
	7	39	0.19	0.53	0.78	1.06
H	1	67	0.14	0.81	1.0	1.02
	2	60	0.18	0.64	0.85	0.87
	7	43	0.19	0.65	0.87	0.87

Table 5.4: Virtual world scale change

5.3.4 Gaze fixation distributions

Raw gaze sequence sample is shown in Fig. 5.12-left. Gaze sequence portions when participant's gaze did not fixate on an object for longer than 100ms were considered involuntary saccades that carry little cognitive load. As the gaze sequence was recorded at average 40Hz, gaze sequences of less than 5 pulses were flagged as saccades, see Fig. 5.12-right.

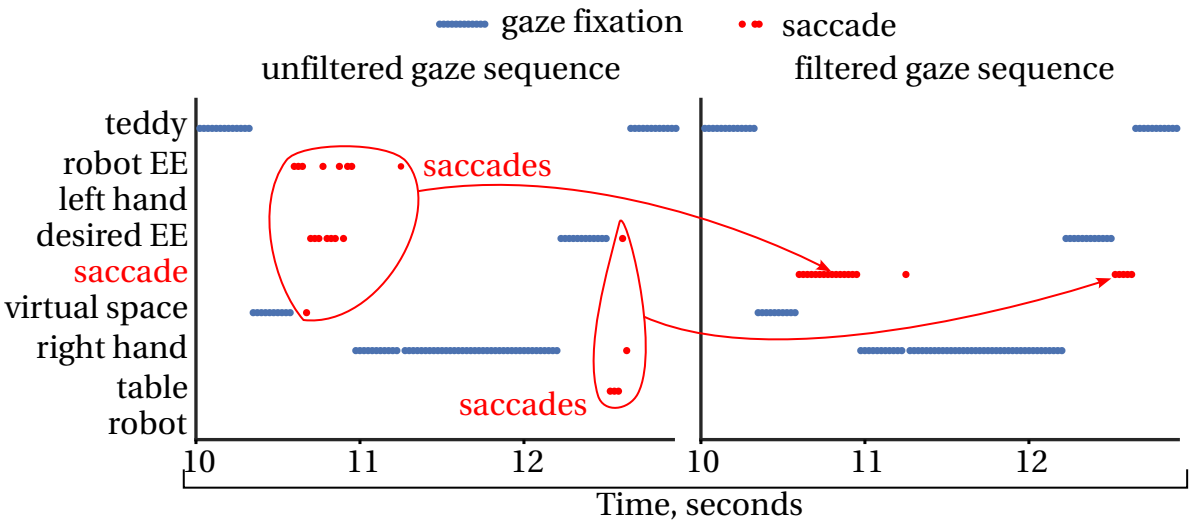


Figure 5.12: Saccades removal sample. Left: unfiltered gaze sequence - gaze sequences of less than 5 pulses (100ms) are considered to be saccades; right: filtered gaze sequence. Blue dot represent fixations, red - saccades

In every test the visual attention of a participant was split by what they were looking at resulting in gaze fixation distribution - fraction of total sample time participant was looking at every gazeable object. Gaze fixation distribution of a randomly chosen sample is given in Fig.5.13.

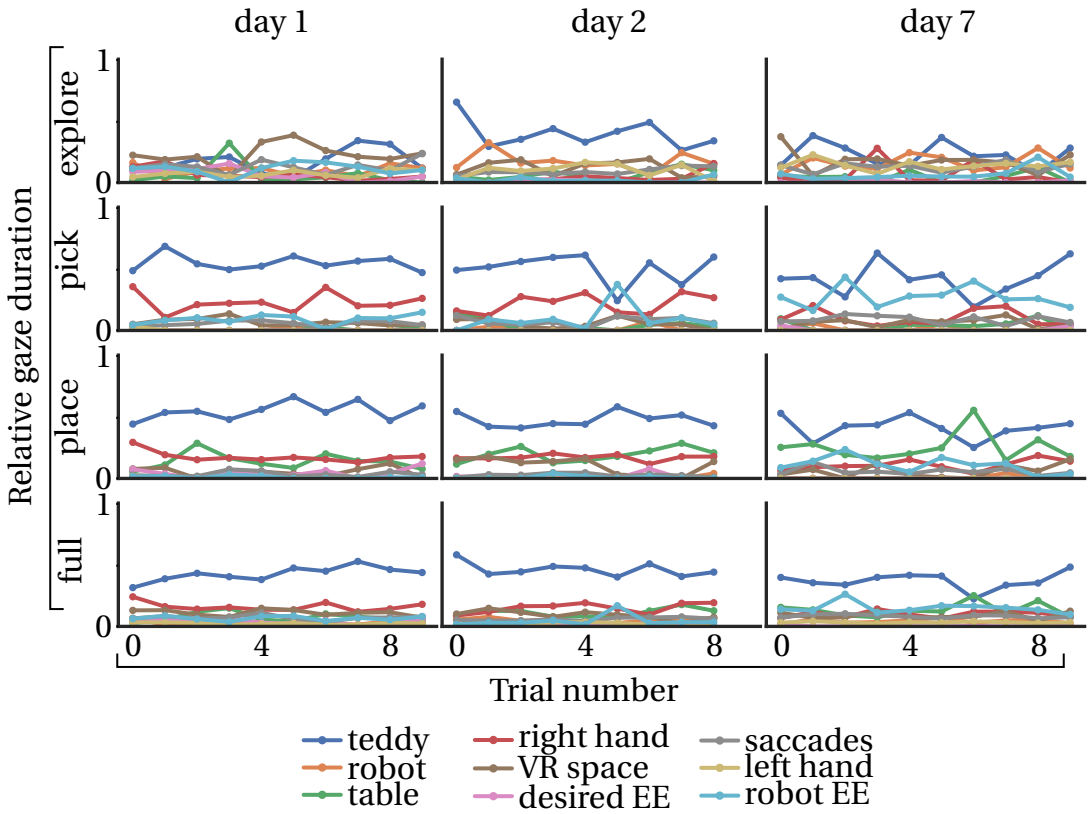


Figure 5.13: Relative gaze durations on tagged objects in VR environment of a random participant split by experiment days and teleoperation phases.

Fig. 5.14 shows the gaze fixation distribution as percentage time gazed at an object averaged across all participants split by experiment day and teleoperation phases. Overall participants gazed statistically significantly most (Tukey HSD test) at the teddy bear, the table and the right hand. Gaze percentages of teddy, table and right hand are also shown in Table 5.5. The gaze distribution is balanced in the exploration stage. In pick and place the teddy, table and the right hand had the most statistically significant fixations. Participants looked very little at the left hand in every phase.

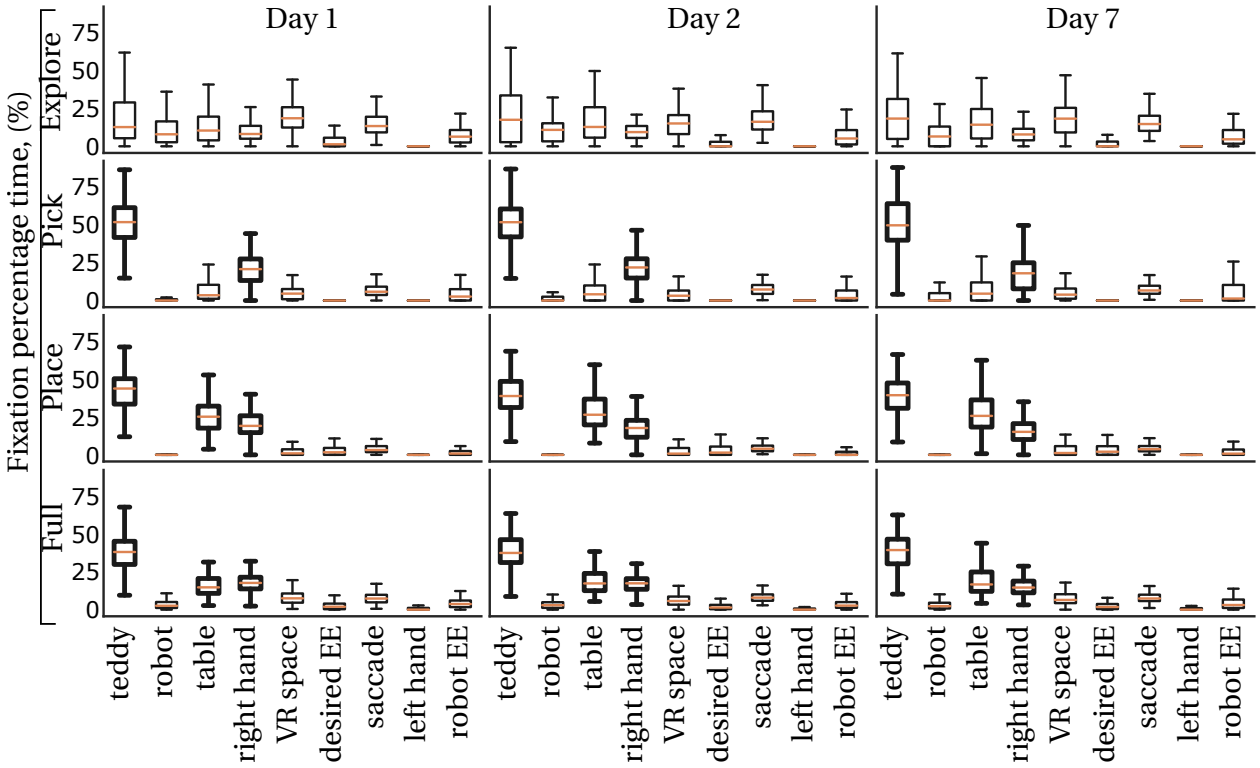


Figure 5.14: Relative gaze fixation across all participants, split by experiment days and teleoperation phases. Bold boxplots indicate objects that had statistically significantly longer fixation times than others, except within themselves, for example in day 1 pick phase right hand fixation time was statistically higher than everything else, except teddy.

Table 5.5: Three most gazed objects' gaze duration percentage (%)

	Day 1				Day 2				Day 7			
Object	expl	pck	plc	full	expl	pck	plc	full	expl	pck	plc	full
Teddy	17	53	43	40	20	54	37	39	16	46	36	35
Table	9	3	23	13	12	5	29	17	14	10	35	22
Right hand	10	24	24	20	12	23	20	19	11	14	13	13

The fixation durations of teddy, table and right hand broken down by participants' videogaming experience is shown in Fig. 5.15. **M** participants start with least teddy fixations on day 1 but participants with no experience end with highest teddy fixations on day 7. **H** participants begin with least table fixations on day 1 but end level with other groups. **N** participants begin with least right hand fixations but end level with **H** participants while **M** participants take the lead.

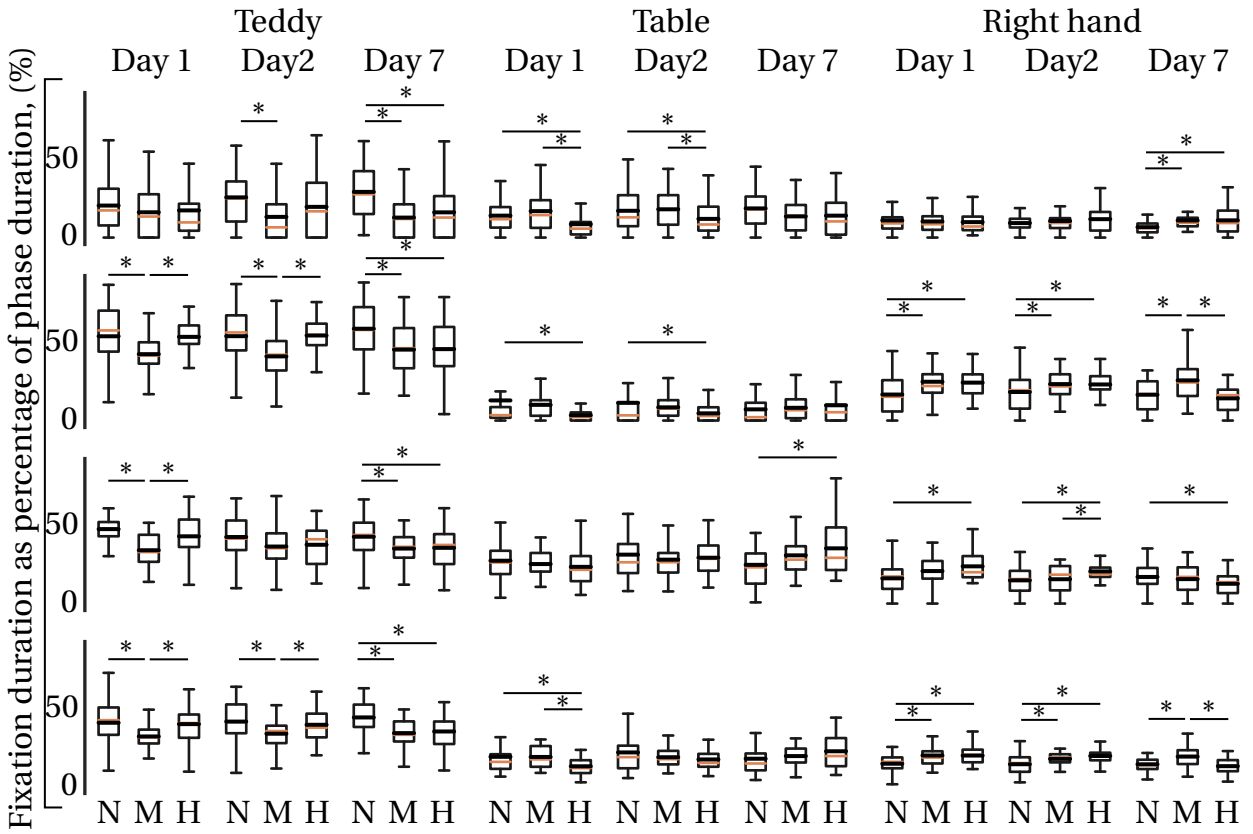


Figure 5.15: Relative gaze fixation comparison across all experiment days and participants' videogaming experience of three most important objects: teddy, table and right hand. *p < 0.05.

The change in visual behaviour within groups on per participant basis was checked similarly to experiment durations with one-way ANOVA with post-hoc Tukey HSD as well as Mann-Kendall test. The condensed summary of tests is shown in Fig. 5.16 (full breakdowns split by participants' videogaming experiences are in Appendix: Fig. A.1 - all participants, Fig. A.2 - **N** participants, Fig. A.3 - **M** participants, Fig. A.4 - **H** participants).

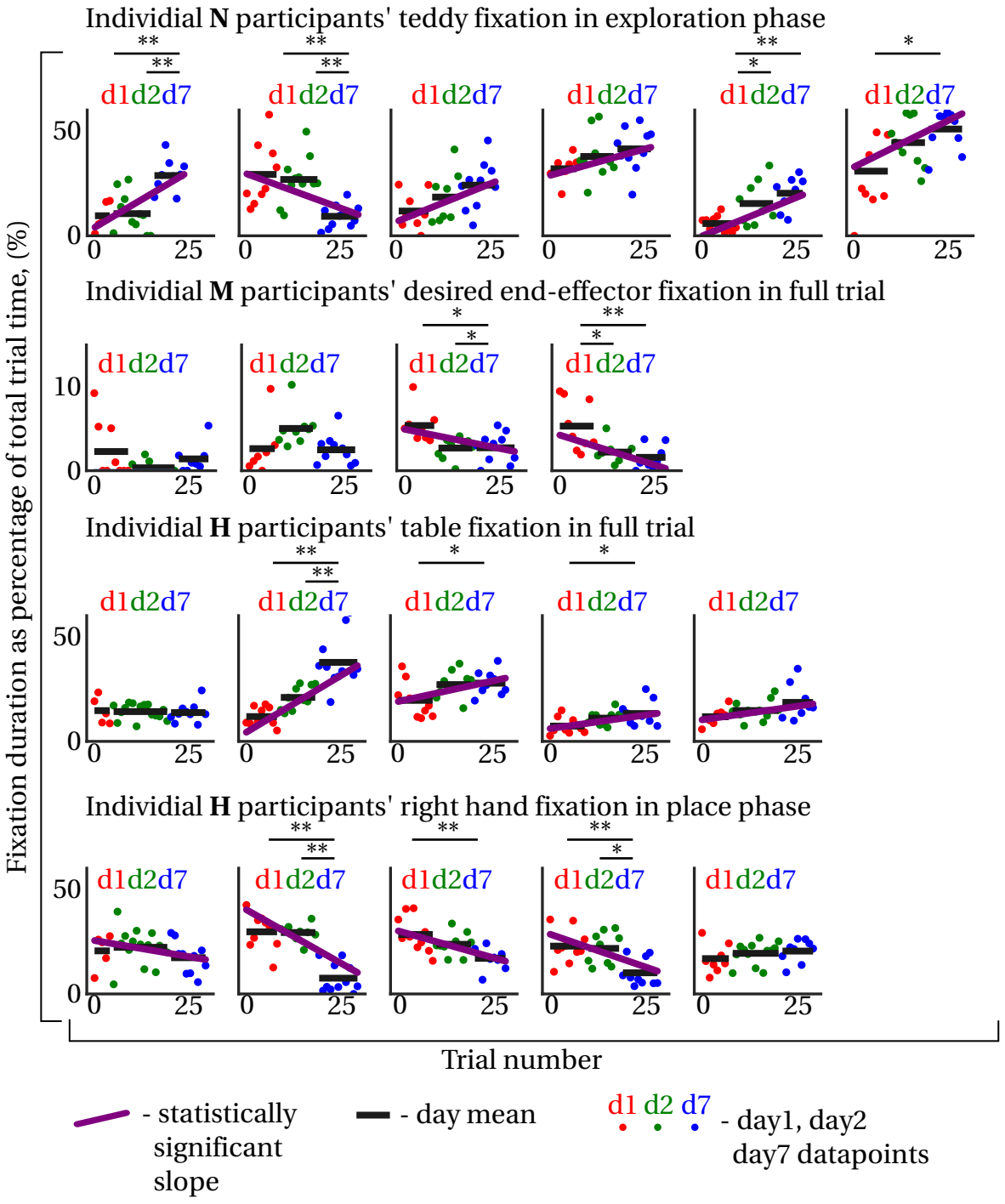


Figure 5.16: Individual participants' fixation duration as percentage of total phase time for objects' with most significantly common visual attention changes per gaming experience group. Color-coded dots represent datapoints of individual days, black lines – corresponding means, purple lines – statistically significant slopes (Mann-Kendall $p < 0.05$), *$p < 0.05$, **$p < 0.01$ Tukey HSD. Note that statistically significant slopes are only displayed for combined day 1 to day 7.

Participants changed their visual attention priority as they trained depending on their prior videogaming experience. It is assumed that any noteworthy visual attention change behaviour is indicated by a corresponding statistically significant fixation slope across all trial days. 5/6 **N** participants showed increasing slope in teddy fixations in combined day 1 to day 7 in the exploration phase. 2/4 **M** participants showed decreasing slope in desired end-effector fixations in full test; we did not observe a common change in right hand fixations. 4/5 **H** participants showed an increasing slope in table fixation in full trial and decreasing slope in right hand fixation in grasping phase.

5.3.5 Gaze shifts and common gaze pairs

A gaze shift was registered whenever a participant shifted their gaze fixation from one uniquely named object to another, excluding saccades. Saccades in between fixations have been removed when registering gaze shifts, for example in Fig. 5.12, from participant has shifted their gaze from "VR space" to the "right hand" and from "desired EE" to the "teddy" through saccades; in both cases saccades were excluded from gaze shifts. A gaze shift matrix of a randomly chosen trial sample is shown in Fig. 5.17 - (participant gazed from columns to rows).

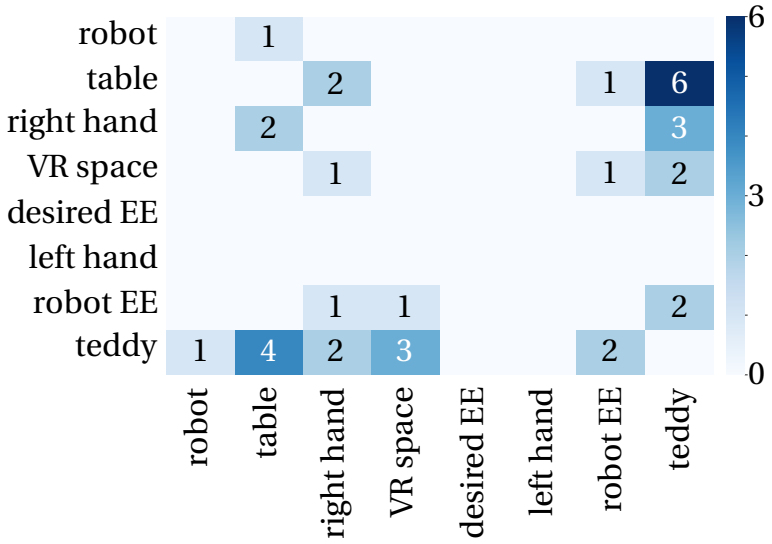


Figure 5.17: Gaze shift matrix of a random trial

An average gaze shift frequency is defied as the average number of gaze shifts per second in a trial. Gaze shift frequencies of three randomly chosen participants are shown

in Fig. 5.18. For majority of participants the frequency of gaze shifts did not change.

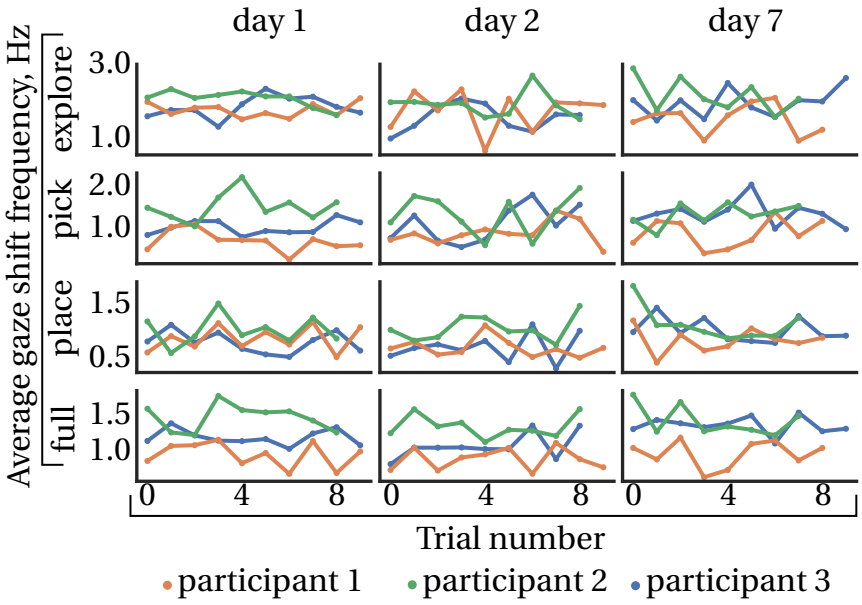


Figure 5.18: Gaze shift frequency curve samples

The importance of objects in separate teleoperation phases was analysed by determining the most common cognitive gaze pairs. Gaze matrices were normalised using the total number of gaze shifts in a trial and gaze shift pairs were generated - for example in the gaze shift matrix in Fig. 5.17 the number of times participant shifted their gaze from "teddy-table" with "table-teddy" is summed up. Next, all gaze pairs that are under 0.90th quantile were removed leaving only the most important gaze pair changes. The quantile value was determined empirically, such that it would consistently leave 4-5 most important gaze pairs per teleoperation phase per participant. Fig. 5.19 demonstrates the most common gaze pairs as they occurred at different teleoperation stages and the number of participants who exhibited those gaze pairs. Neither ANOVA with post-hoc Tukey HSD nor Mann-Kendall have shown statistically significant changes in these important gaze pair proportions. Furthermore, no significant differences in gaze pairs based on participants' prior videogame experience was observed with the exception that gamer participants tended to look less at the desired end-effector.

Participants did not often look at certain objects in different phases of teleoperation. Neither the robot's body nor the left hand were part of most common gaze shifts overall. Right hand was not part of common gaze shifts in exploration phase. Desired end-

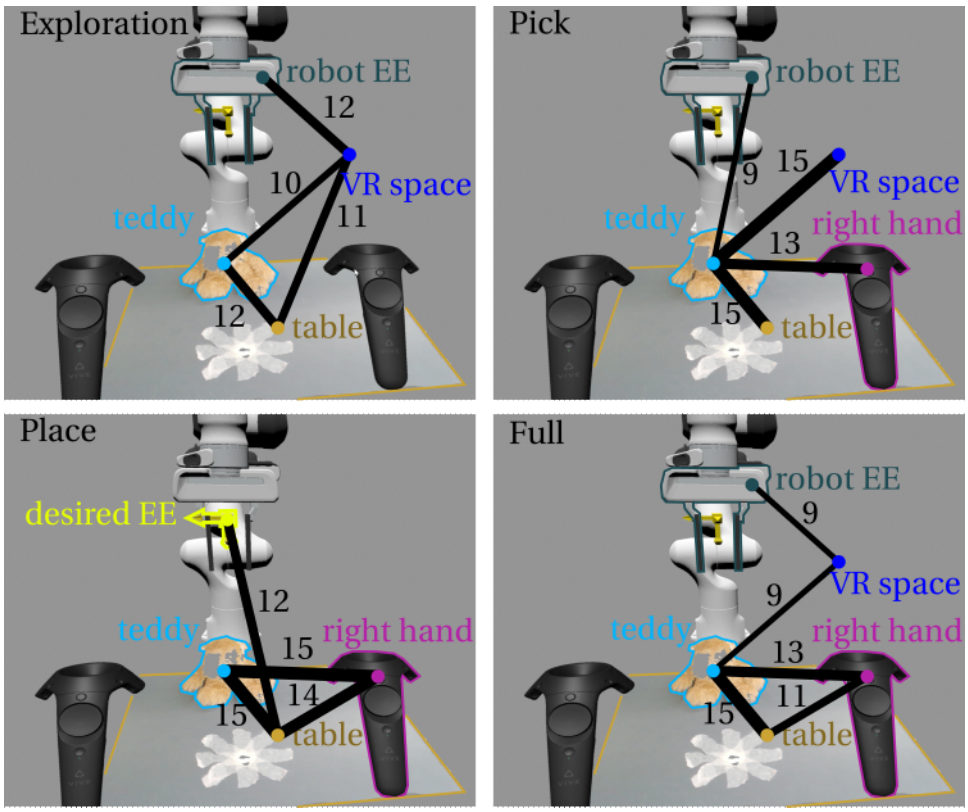


Figure 5.19: Most common gaze pairs and number participants that exhibited these gaze pairs across teleoperation phases

effector was only part of common gaze shifts in place phase. VR space and the robot's end-effector were not present in common gaze shifts in place phase.

There are also differences in what gaze pairs are present in different teleoperation phases. Participants' gazes most commonly shifted between the teddy and the table – 15/15 in pick, place and full. Participants only shifted their gazes between robot's end-effector to VR space and between VR space and the table in the exploration phase. In both pick and place phases participants shifted their gazes between the teddy and the right hand, but participants did not shift their gaze between the right hand and the table in the pick phase.

5.4 Discussion

5.4.1 Task execution time

Given the simplicity of the experimental task it was expected that all participants would perform the task faster on subsequent days and be proficient in performing the task by the end of the last day of training. Statistically significant time improvement in full trials for the majority of participants was observed between days 1 and 2 as well as days 1 and 7. Very few participants have showed statistically significant monotonous slopes in days 1 or 2. However when day 1 and day 2 are combined statistically significant slopes were observed that correspond to time improvements reported above. The variances in execution times were larger on day 1 than day 2. It can be argued that the variances in day 1 trials were too large to detect learning slopes; then on day 2 variances reduced but there was little actual slope. Hence, it can be concluded that most of the learning happened on day 1.

Participants showed the most consistent time improvement in the exploration phase which was also the most physically and cognitively demanding teleoperation phase as reported by participants. In exploration phase participants had to navigate the virtual reconstruction of the remote environment by translating, rotating, scaling the virtual world using hand gestures. The exploration phase was considered finished once participants have interacted with the desired end-effector mesh. In early trials the experimenter observed that many participants failed to judge distances and the scale of the virtual world, for example: participants would place themselves outside of an arm's reach away from the robot and could not start the next teleoperation phase. By com-

parison, grasping and manipulation required fewer physical actions leaving less space for improvement.

Participants' prior video game experience affected how quickly participants perform the experimental task and how much they improved day to day. Participants with no prior video gaming experience (**N**) performed the task slower than participants with video game experience (**M, N**) on every experiment day, although the gap between the groups decreased overtime - **N** participants exhibited the largest time improvement particularly in the exploration stage. In addition to training challenges discussed above, **N** participants have shown to be less comfortable with handheld controllers, taking longer to remember which button to press and how to press them. There was no statistically significant difference in task completion time between **M** and **H** participants. But **H** participants tended to perform more uniformly as a group; they also exhibit less variation in task completion times than **M** participants. Furthermore, **H** participants were the only group, with all members uniformly demonstrating statistically significant time improvement.

5.4.2 Virtual-to-real remote environment reconstruction scale

As participants started to execute the task faster they tended to reduce their virtual world scale and the number of times they adjusted it per trial. However, when examined individually only 7 out of 15 participants have shown statistically significant reductions in virtual world scale between experiment days - i.e. the behaviour is not uniform. Hence, it can be concluded that for a robotic pick-and-place task with large grasp and place tolerances as in the presented experiment, expert teleoperators are likely but not guaranteed to use a smaller virtual world scale compared to a novice teleoperator; furthermore, it is expected that overall expert participants are also likely to use different (subject to task accuracy) virtual world scales than novice teleoperators.

Participants with video gaming experience (**M, H**) used a smaller virtual world scale compared to participants with no experience (**N**). Surprisingly, no considerable differences between medium (**M**) and high (**H**) experience participants was observed, except for the number of average scale changes and their durations - **M** participants tended to do fewer slower scale changes than **H** participants. It is unlikely that gamers will use a smaller virtual world scale than non-gamers in every teleoperation task. However, it can be concluded that the video gaming experience indeed helps operators to determine a

more optimal virtual world scale and navigate to it more efficiently.

5.4.3 Gaze distribution

The visual attention of human-operator during VR-based robotic pick and place task largely corresponds to human real world pick and place task [174]: there is high visual attention on target object, object area (pick and place locations) with exception of higher focus on the right hand, which is used differently in robotics task to begin with. The robot's body was present in the exploration phase fixations but was not part of the common gaze pairs. The robot's end-effector and desired end-effector mesh by comparison were more present both in all phases' fixations and common gaze pairs. It can be argued that the robot's body visualisation is not necessary during pick and place phases, it is sufficient to only render the robot's end-effector and the desired end-effector mesh. All participants looked very little on the left hand. In the VR-based robot teleoperation framework the left hand is only used in gesture-based navigation. It is concluded that participants either navigated very quickly and/or didn't have to look at their hands when performing the gesture-based navigation. In the latter case it can be argued that the left hand does not need to be visually rendered. The significance of VR space fixations and what cognitive process they correspond to is unclear – it could be argued that participants could have used the VR space as gaze resting space.

Participants changed their visual attention priorities as they learned to teleoperate the robot. These changes differed for participants based on their prior video gaming experience. Interestingly, fixation distribution across groups did not converge as participants in different groups ended their day 7 with different fixation distributions. It could be argued differences in task execution time are connected to these different fixation distributions. **N**-group participants increased teddy fixation in the exploration phase. We did not observe corresponding statistically common trends of participants looking less at other objects. Therefore, it can be concluded that as **N** participants started to value the teddy more during the exploration, they de-valued every other object in the virtual world equally. The reason for this behaviour is yet unclear. Participants with medium gaming experience (**M**) de-valued the desired end-effector overall. This is interesting given that for high experience participants (**H**) the desired end-effector was least present in their gaze pairs. It could be argued that the **M** participants' behaviour converged to **H** group's when looking at the desired end-effector. Participants with high

gaming experience increased the table fixation time in full trial and decreased right hand fixation time in the pick phase. It can be argued that as they performed the task faster they were left with relatively more downtime when robot would execute the trajectory, during which participant's gaze would rest on the largest objects in the scene, which are the robot's body and the table. Given that participants looked at the robot very little it can be concluded that the framework has high robot trust as participants did not inspect robot's motions for potential collisions.

Common gaze pairs demonstrate object priority links during different phases of teleoperation. In exploration phase the gaze shifted notably in teddy-table-VR space triangle, in place phase – teddy-table-right hand triangle, meanwhile in pick phase gaze shifts centered around the teddy. This finding can be used for designing visual cues that could help novice operators, for example communicate more important object's in VR interface in corresponding teleoperation phases by manipulating the opacity of objects.

5.5 Limitations and future work

A number of trials were excluded from the analysis as outliers post-experiment and the number of trials per day per participant was not constant. Although, the total time participants have spent on the experimental setup is similar, i.e. up to 40 minutes per day, this imbalance could have affected results, for example: day 1 had the most outliers and failed trials resulting in lower confidence in Mann-Kendall test.

One common issue observed by the experimenter and reported by participants (especially non-gamer participants) was confusing/unresponsive buttons for closing and opening the gripper on the circular trackpad. Indeed, according to experiment events logs participants would often press close/open gripper repeatedly in grasping and manipulation phases. For example: participant could attempt to open the gripper while it was already open, realise the mistake and attempt to close the gripper; or participant would press on the areas on the circular trackpad that were not associated with the gripper – reported as unresponsive. In both cases participants would spend more time grasping/releasing the teddy than necessary resulting in higher variance in pick and place phases' execution times. The unresponsiveness part of the issue could have been remedied by assigning gripper open/close to physical buttons instead of trackpad. The button confusion was more common amongst non-gamer participants which was expected as they are less familiar with gaming devices.

It should also be noted that the experiment was designed to be simple and had large tolerance for grasping and placing accuracies which can be considered a limitation for this study. In a different experiment, for example, a cup-stacking experiment in which grasping and placing accuracies need to be much stricter, participants might have had to increase their virtual world scale (zoom-in) more. It can however be argued that even there, expert participants would use different virtual world scales.

Participants have rated themselves on a scale of 1 – 10 on how often they played video games, which in retrospective could be ambiguous to quantify and is a poor indicator of their relevant skills. This can also be considered a limitation of the study. In future work, a more rigorous way to evaluate and quantify each participant's level of experience is recommended. For example, participants could perform a separate test designed solely to assess their ability and give them a rating based on their performance in addition to having them fill out a questionnaire.

Participants were not incentivised nor dissuaded from placing the teddy accurately on the placement mark. Some participants tended to value accuracy more and spent more time placing the teddy. This increased variation in the manipulation phase execution time. However, teddy placement positions were recorded, and the accuracy precision of placement against participants prior experience can be analysed in the future work.

Gaze registration was less accurate on the desired end-effector. Whenever participants had moved the desired end-effector, the mesh of the right hand would partially occlude/overlap the one of desired end-effector. Hence there could be some confusion whether participants were looking at the right hand or the desired end-effector. Similarly, the teddy also occluded/overlapped the desired end-effector when grasp position was set.

OctoMap was used as a rough meshing solution for the point cloud as they required collision meshes for gaze registration. This could present two potential sources of inaccuracy in gaze registration. First, OctoMap bounding space around the point cloud was slightly larger than point clouds, hence some gazes may have been registered incorrectly. Second, although OctoMap was tuned to "forget" currently unobserved nodes, some gazes could have been registered on parts of the map that were meant to be forgotten - for example on the old position of the manipulation object, while manipulation object was moving. In future works, it is recommended to use more real-time robust meshing solution for point cloud.

It is important to consider the general accuracy of Tobii's gaze-to-object-mapping, particularly the absence of confidence scores for the most likely gaze candidate, which was a limitation of the middleware used in the experiment. However, the manipulation object, area, and hand had such a large lead in gaze distribution that their lead is unlikely to have been caused solely by classification inaccuracies. Additionally, since the most likely candidate was double-checked against gaze rays, our results are likely a conservative estimate. It should also be noted that blinking and pupil information were not recorded, which raises the question of whether some saccadic classifications may have been affected by blinking or unsuccessful tracking. To address this issue, future work could consider double-checking saccade classification against blinking history. Overall, while there are limitations to our study, our results suggest that the manipulation object, area, and hand have a significant impact on gaze behavior and should be considered in the design of user interfaces.

5.6 Chapter conclusions

This chapter presented a study on how operators learn to use VR-based teleoperation and their progression from novice to expert operators as well as how their visual preferences shift and how participants' prior experiences affect the previous two questions.

Experimental results demonstrate that for a simple pick and place task in supervised control mode operators do most of the learning within the first eight to ten trials. When teleoperation is broken down into exploration, pick and place phases, most of the learning happened in the exploration phase.

Participants' prior video gaming experience affected how quickly participants perform the experimental task and their rate of improvement. Participants with no prior video gaming experience performed the task consistently slower than participants with video gaming throughout the experiment. It should however be noted that the gap between participant groups decreased over time as non-gamer participants also exhibited the most learning. Participants with high gaming experience performed more uniformly as a group and exhibited less variation.

By the end of training operators used a smaller scale (although not every individual operator) which can contribute towards shorter task duration time. Given that the task had large tolerances for grasping and placing, the smaller virtual world scale allowed operators to perform the task faster as it required smaller physical movements. As ex-

pected operators with prior video gaming experience did perform the task faster. In future work, it would be interesting to examine operators' behaviours in different tasks, such that task-specific optimal virtual world scales can be derived and applied automatically.

Overall the visual attention of operators during VR-based robotic pick and place task largely corresponds to human real world pick and place task: there is clearly high visual attention on the target object and object area (pick and place locations). Unlike real world pick and place, operators exhibited higher focus on their dominant hand, which has a different function in the robotic task. Participants' gaze distribution changed differently based on their video gaming experience. Participants changed their visual attention priorities as they learned to teleoperate the robot. These changes differed for participants based on their prior video gaming experience. Interestingly, fixation distribution across groups did not converge to a single distribution pattern.

Participants looked little at the non-dominant hand overall and the robot's body in pick and place phases. It can be argued that the framework has high robot trust as participants did not inspect the robot's motions for potential collisions. Furthermore, it can be argued that the non-dominant hand and robot's body can be rendered with less graphical fidelity.

Most common gaze pair patterns were identified for every teleoperation phase. This can be used for designing visual cues that could help novice operators, for example, communicate the more important object's in the VR interface in corresponding teleoperation phases by manipulating the opacity of objects.

6 Conclusions

This thesis presents research on VR-based robot teleoperation. As demonstrated by prior research [20, 1] in comparison to conventional robot teleoperation interfaces, virtual reality (VR)-based interfaces provide a human-operator with improved spatial perception, more intuitive control and remote environment exploration enabling challenging telerobotics applications like disaster relief, surgery, and remote exploration. However, as VR-based telerobotics is a relatively new field it is yet unclear how traditional teleoperation techniques for exploration/visualisation of the remote environment and robot control are best transferred and evolved in VR human-robot interfaces. In particular, this work aimed to improve remote environment exploration and visualisation, effects of the virtual-to-real scale of remote environment reconstruction in the virtual reality interface on the human-operator's ability to control the robot and human-operator visual attention patterns during robot teleoperation with virtual reality interface. Below is a discussion of how these aims were achieved.

In order to perform the VR-based robot teleoperation studies presented in this thesis a VR-based robot teleoperation framework was developed. The framework builds upon previous works [4, 30] and expands upon them in multiple ways. The framework is hardware agnostic, making it compatible with various robotic systems and cameras. Real-time direct teleoperation and supervised control can be achieved with any ROS-compatible robot while visualising the environment through any ROS-compatible RGB and RGBD cameras. The framework also includes mapping and segmentation of the remote environment. Tactile exploration is used to determine an object's materials that are communicated visually to the operator. The framework allows navigation and control through any Unity-compatible VR headset and controllers or haptic devices, with a consistent set of gestures and functionalities for operator ease of use. Furthermore, navigating the VR space is designed to be non-physically demanding and not induce motion sickness, allowing for extended use. In future work, the presented framework could benefit from improved mapping, meshing and segmentation of the remote envi-

ronment as it would increase the accuracy of gaze tracking and improve the applicability of the framework.

As mentioned above the first goal was to improve the state of remote environment visualisation in VR-based robot teleoperation. Point clouds are the primary means of visualising unstructured remote environments in 3D. However, in the current state of technologies point clouds often suffer from distortions, and occlusions and do little to represent objects' texture. If objects in the remote environment are inaccurately represented in the VR reconstruction, the operator can make incorrect judgments about their nature and/or shape, leading to poor decision-making during teleoperation. To alleviate this problem two studies were performed. The visual exploration participant study has shown that end-effector mounted RGBD camera with OctoMap mapping of the remote environment allows the operator to explore the remote environment with less point cloud distortions and occlusions compared to [4, 158, 130] whilst using a relatively small bandwidth. Unlike [61, 62, 3] this approach fits well for unstructured remote environments. The tactile exploration study aimed to further address the challenges of point cloud visualisation by providing the operator with information about the objects' materials. A novel method for classifying objects' materials and presenting this information visually in the VR interface was proposed in order to improve the operator's decision-making suitable for the exploration of hazardous environments [83]. For future work, it is advised to expand tactile classifiers to estimate objects' hardness/roughness rather than assigning hard-coded classes. Furthermore one can consider real-time classification instead of post-scan classification.

Unlike the real world, in VR reconstruction the remote environment can be scaled up or down. The effects of virtual world dynamic scaling on teleoperation flow are not yet fully understood. Two studies have been performed on the effect of the virtual-to-real scale of the remote environment reconstruction on the operator's ability to control the robot. The first study investigated the rate mode control with constant and variable mapping of the operator's joystick position to the speed (rate) of the robot's end-effector. The variable mapping depended on the virtual world scale. The study demonstrated how the rate mode control and variable scaling based on the VR reconstruction scale can be efficiently used for seated VR-based robot teleoperation when the operator's arms are supported to reduce tiredness. The corresponding participant study shows that variable mapping allowed participants to teleoperate the robot more effectively, by adjusting the VR visual scale albeit at a cost of increased perceived work-

load. To the best of the author's knowledge, no comparable works exist: although rate mode is a fairly well-studied field [238, 233] and active virtual world scaling was proposed parallel to this thesis in [66] it was never studied in the context of VR-based robot teleoperation. The second study explored how operators used a virtual world scale in supervised control. The virtual world scale used by participants was compared at the beginning of a 3-day experiment when they were considered to be novices and at the end of it when they were considered to be experts. In pick-and-place robotic task expert teleoperators as a group used a smaller virtual world scale than novices, although this behaviour was not exhibited by every teleoperator individually. The study also demonstrated that participants' prior video gaming experience affects the virtual world scale as participants with video gaming experience used smaller virtual world scales and used fewer scale changes. This study reinforces the idea that the video gaming experience is beneficial for robot teleoperation [236, 237]. In future work, one can consider performing similar virtual world scale studies to determine the optimal scale for different teleoperation tasks and phases, such that they can be applied automatically.

Similar to the virtual reconstruction scale study the visual attention data of operators as they learned to teleoperate the robot using our VR-based teleoperation framework was analysed. The results revealed the most important objects in the VR reconstructed remote environment as indicated by operators' visual attention behaviour as well as how operators' visual attention behaviour changed as operators got better at teleoperating the robot and how operators' prior experience affect their ability to teleoperate the robot. The visual attention of the operator during VR-based robotic pick and place task largely corresponds to human real world pick and place task [174]: there is high visual attention on the target object, object area and manipulation hand. Hence it can be hypothesised further that the knowledge from real world tasks can be transferred to VR-based robot teleoperation to further improve VR-interfaces. Understanding human-operator's visual attention patterns and their dependence on their prior experience can be used to further improve VR-based robot teleoperation interfaces. In future work, it is advised to utilise the findings of this study in order to automatically adjust the visibility of high and low-priority objects.

To conclude research presented in this thesis made several contributions to the field of VR-based robot teleoperation. Firstly a novel VR-based robot teleoperation framework was developed, which provides a flexible and intuitive interface for controlling remote robots using VR technology. Secondly, methods for remote environment vi-

sual reconstruction and exploration in VR, which can enhance the operator's situational awareness and improve their ability to navigate and control the robot were investigated. Thirdly, a novel method for classifying objects' materials and presenting this information visually in the VR interface was proposed. This method can improve the operator's ability to identify and interact with different objects in a remote environment. Fourthly, the effects of the virtual world scale on the operator's ability to teleoperate the robot in different control modes were investigated. The findings can inform the design of more effective VR-based teleoperation systems. Finally, the operator's visual priorities during VR-based robot teleoperation and their dependency on the operator's experiences were investigated. This research can provide insights into operator training and the design of more effective VR-based teleoperation systems.

A Supplementary materials

A.1 Visual attention extra figures

This four figures present participants' gaze priority changes in the visual attention study presented in chapter 5. Figures present the number of participants that have shown statistically significant increase/decrease in relative fixation time on objects during robot teleoperation using the VR framework presented in chapter 2. Change in relative fixation time determined by statistically significant change in relative gaze fixation means using Tukey HSD test and presence of corresponding statistically significant slopes using Mann-Kendall test. Figures were omitted from the chapter due to their size.

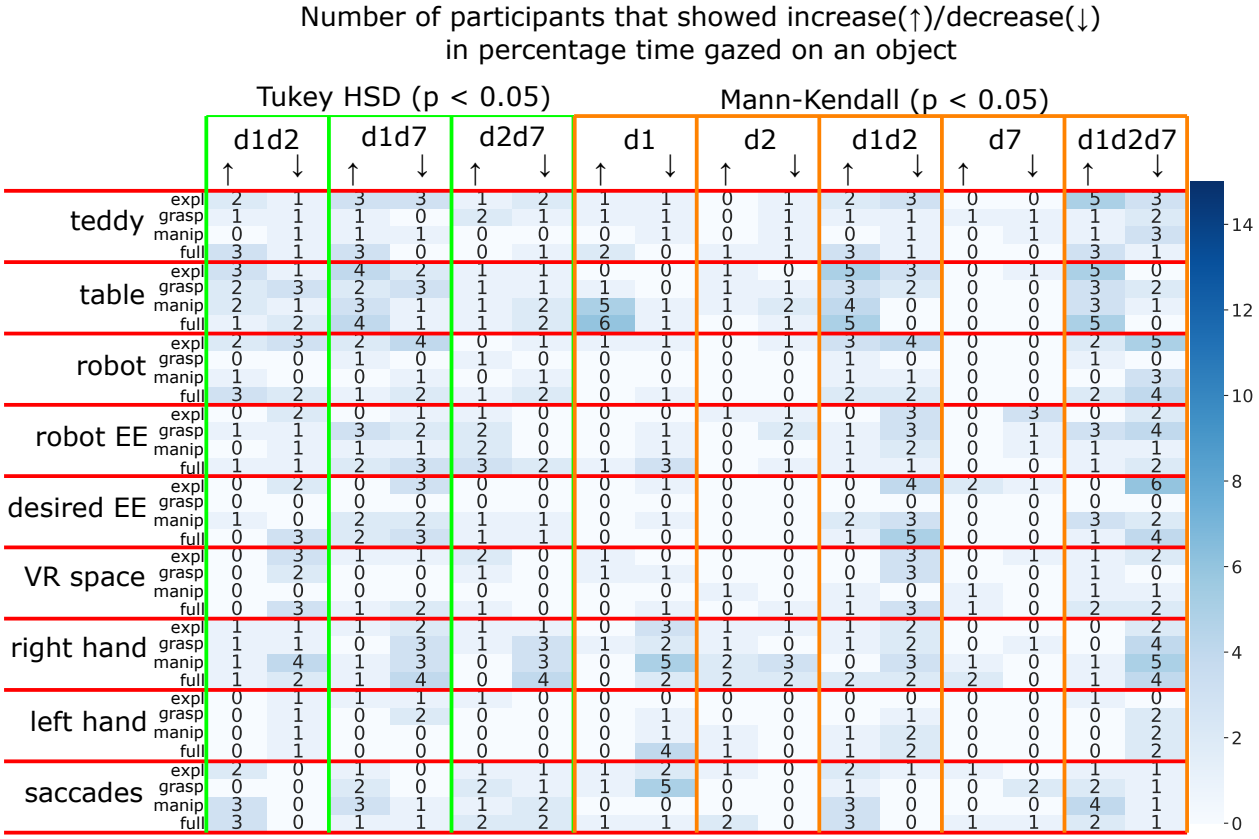


Figure A.1: All participants' gaze priority changes

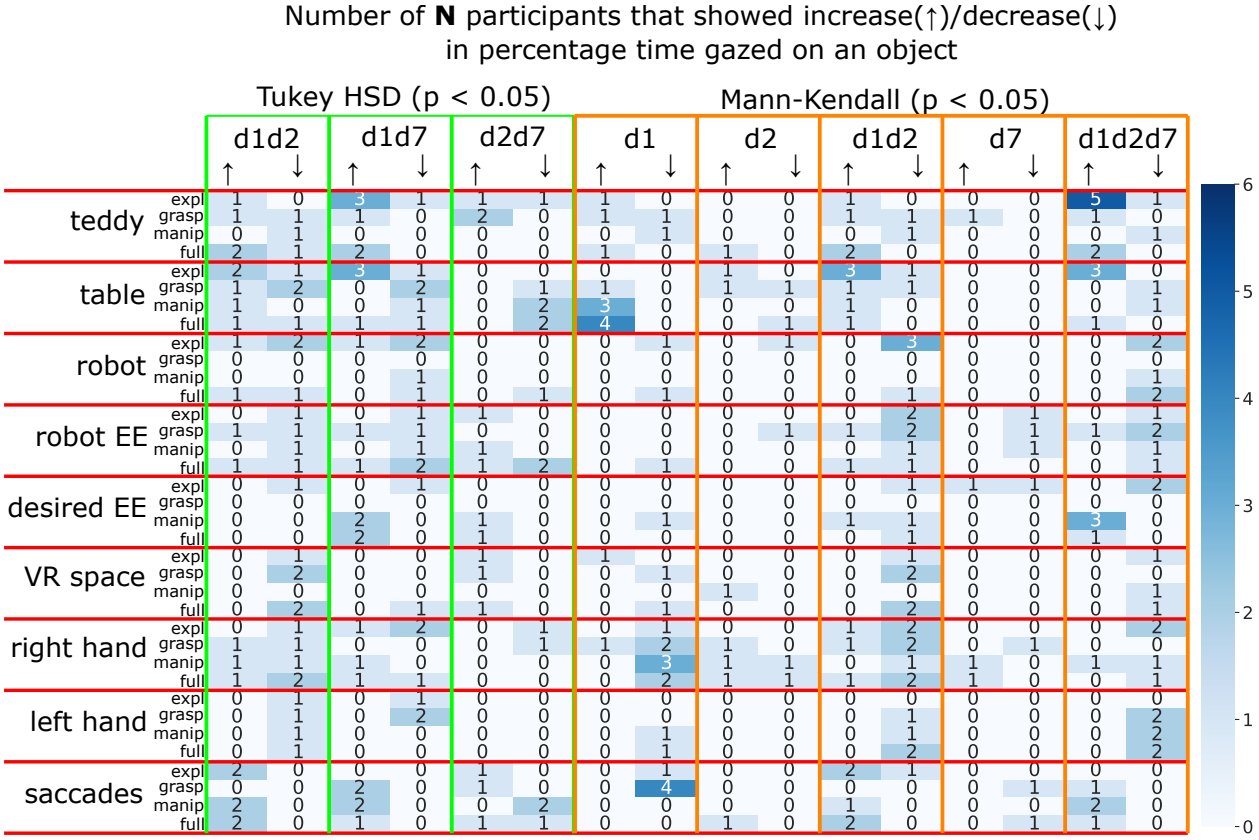


Figure A.2: No gaming experience participants' gaze priority changes

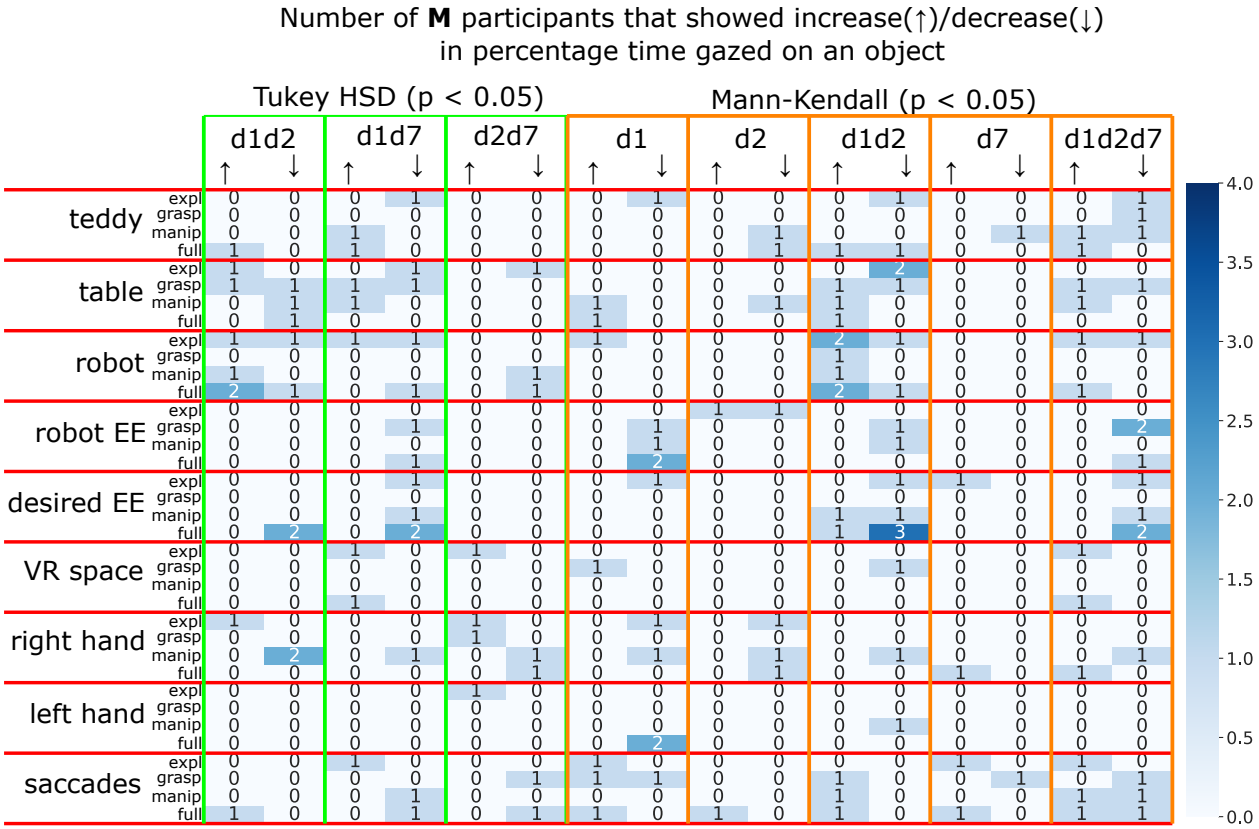


Figure A.3: Medium gaming experience participants' gaze priority changes

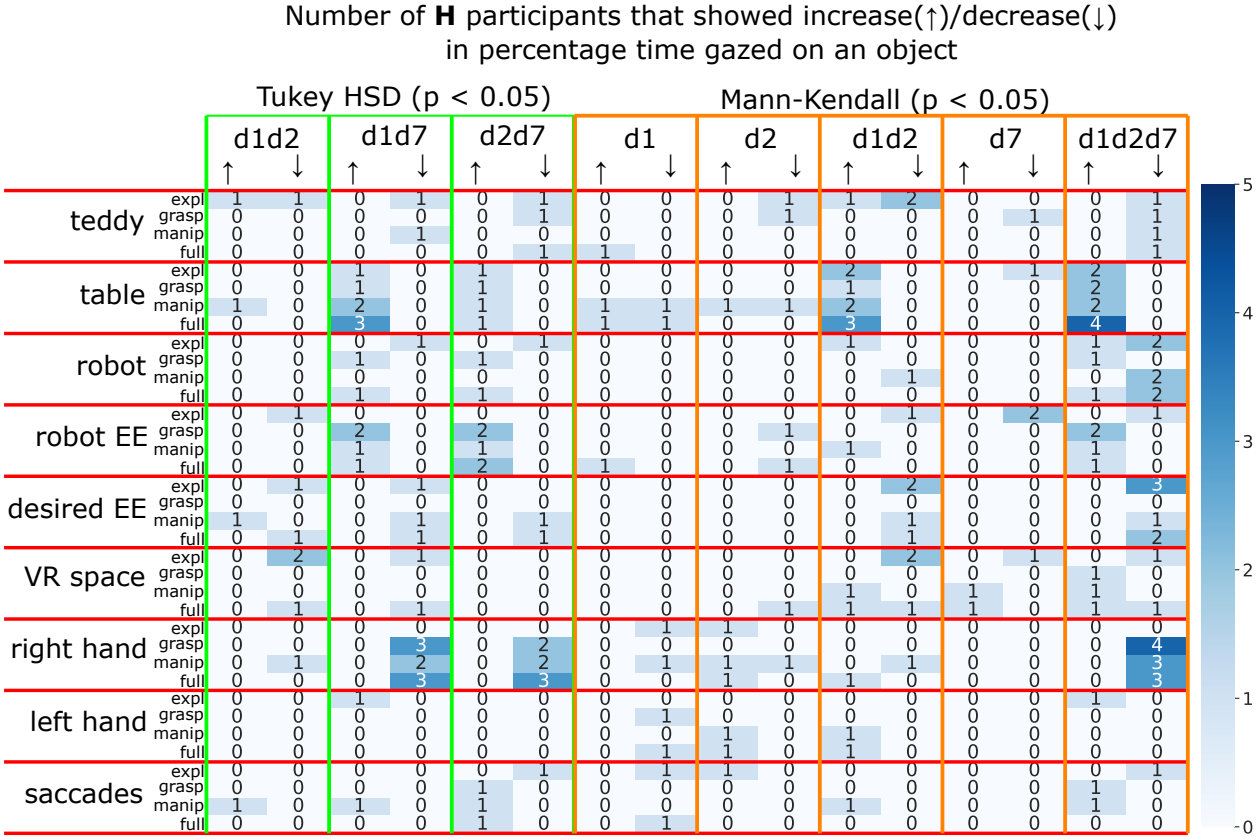


Figure A.4: High gaming experience participants' gaze priority changes

A.2 Virtual world scale and visual attention questionnaire

This is a questionnaire for "Visual Attention in Virtual Reality Based Robot Teleoperation" study. All information provided by participants is deanonymized and stored securely for the duration of the study and disposed afterwards.

(Day 0 only) Please indicate your previous experience with VR from 1 (I have never used VR before) to 10 (I use VR on a daily basis):

_____

(Day 0 only) Please indicate your previous experience with robot teleoperation from 1 (I have no experience teleoperating a robot) to 10 (I teleoperate robots daily):

_____

(Day 0 only) Please indicate your previous experience with videogames from 1 (I do not play videogames) to 10 (I play videogames daily):

_____

(Day 0 only) Please indicate your eyesight condition:

- normal
- corrected to normal with lenses
- corrected to normal with glasses but I can perform routine activities comfortably without glasses.

Please indicate which part of robot teleoperation procedure in VR you found **cognitively** most demanding:

- adjusting the view of remote environment (robot and objects) in VR using gestures;
- setting the desired robot's position by moving the corresponding axis mesh,
- estimating the validity of grasp from point cloud visualization of the remote environment,
- planning robot's motions for grasp and manipulation.
- other, please specify: _____

Please indicate which part of robot teleoperation procedure in VR you found **physically** most demanding:

- adjusting the view of remote environment (robot and objects) in VR using gestures;
- setting the desired robot's position by moving the corresponding axis mesh,
- estimating the validity of grasp from point cloud visualization of the remote environment,
- planning robot's motions for picking and placing.
- other, please specify: _____

Please indicate how well you think you performed on the experimental task from 1 (I did poorly) to 10 (I did great):

_____

(Day 1 and 7 tests only) Please indicate how your performance compare to your previous day of testing from 1 (I did not improve at all) to 10 (I performed much better):

_____

Bibliography

[1] Wai-keung Fung, Wang-tai Lo, Yun-hui Liu, and Ning Xi, "A case study of 3D stereoscopic vs. 2D monoscopic tele-reality in real-time dexterous teleoperation," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 181–186, IEEE, 2005.

[2] D. Whitney, E. Rosen, E. Phillips, G. Konidaris, and S. Tellex, "Comparing Robot Grasping Teleoperation across Desktop and Virtual Reality with ROS Reality," *International Symposium on Robotics Research*, pp. 1–16, 2017.

[3] S. Kohn, A. Blank, D. Puljiz, L. Zenkel, O. Bieber, B. Hein, and J. Franke, "Towards a Real-Time Environment Reconstruction for VR-Based Teleoperation Through Model Segmentation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9, IEEE, oct 2018.

[4] D. Whitney, E. Rosen, D. Ullman, E. Phillips, and S. Tellex, "ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, no. July, pp. 1–9, IEEE, oct 2018.

[5] B. Omarali, F. Palermo, K. Althoefer, M. Valle, and I. Farkhatdinov, "Tactile Classification of Object Materials for Virtual Reality based Robot Teleoperation," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 9288–9294, 2022.

[6] B. Omarali, K. Althoefer, F. Mastrogiovanni, M. Valle, and I. Farkhatdinov, "Workspace Scaling and Rate Mode Control for Virtual Reality based Robot Teleoperation," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pp. 607–612, 2021.

[7] I. Vitanov, I. Farkhatdinov, B. Denoun, F. Palermo, A. Otaran, J. Brown, B. Omarali, T. Abrar, M. Hansard, C. Oh, S. Poslad, C. Liu, H. Godaba, K. Zhang, L. Jamone, and K. Althoefer, "A suite of robotic solutions for nuclear waste decommissioning," *Robotics*, vol. 10, no. 4, pp. 1–20, 2021.

[8] B. Omarali, B. Denoun, K. Althoefer, L. Jamone, M. Valle, and I. Farkhatdinov, "Virtual Reality based Telerobotics Framework with Depth Cameras," *29th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2020*, pp. 1217–1222, 2020.

[9] B. Omarali, F. Palermo, M. Valle, S. Poslad, K. Althoefer, and I. Farkhatdinov, "Position and Velocity Control for Telemanipulation with Interoperability Protocol," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11649 LNAI, pp. 316–324, 2019.

[10] R. C. Goertz, "Master-Slave Manipulator," tech. rep., Argonne National Laboratory (ANL), Argonne, IL (United States), mar 1949.

[11] R. S. Mosher, "Handyman to Hardiman," *SAE Technical Papers*, vol. 76, pp. 588–597, 1967.

[12] S. Chutia, N. M. Kakoty, and D. Deka, "A review of underwater robotics, navigation, sensing techniques and applications," *ACM International Conference Proceeding Series*, vol. Part F1320, pp. 1–6, 2017.

[13] W. R. Ferrell, "Remote manipulation with transmission delay," *IEEE Transactions on Human Factors in Electronics*, vol. HFE-6, no. 1, pp. 24–32, 2013.

[14] W. R. Ferrell, "Adaptive Supervisory Control of Remote Manipulation.," *Proceedings of the IEEE Conference on Decision and Control*, no. 2, pp. 549–552, 1977.

[15] S. G. Weinbaum, *Pygmalion's Spectacles*. Wonder, 1935.

[16] M. L. Heilig, "Sensorama simulator," *US PAT. 3,050,870*, 1962.

[17] C. Comeau, "Headsight television system provides remote surveillance," *Electronics*, pp. 86–90, 1961.

[18] H. McLellan, "Virtual realities," *Handbook of research for educational communications and technology*, pp. 457–487, 1996.

[19] S. S. Fisher, E. M. Wenzel, C. Coler, and M. W. McGreevy, "Virtual interface environment workstations," in *Proceedings of the Human Factors Society Annual Meeting*, vol. 32, pp. 91–95, SAGE Publications Sage CA: Los Angeles, CA, 1988.

[20] A. Kron, G. Schmidt, B. Petzold, M. I. Zäh, P. Hinterseer, and E. Steinbach, "Disposal of explosive ordnances by use of a bimanual haptic telepresence system," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2004, no. 2, pp. 1968–1973, 2004.

[21] M. E. Moran, "The da vinci robot," *Journal of endourology*, vol. 20, no. 12, pp. 986–990, 2006.

[22] M. Sobhani, M. Giuliani, A. Smith, and A. G. Pipe, "Robot Teleoperation through Virtual Reality Interfaces: Comparing Effects of Fixed and Moving Cameras," *Vam-Hri*, vol. 2657, pp. 1–9, 2020.

[23] G. H. Ballantyne and F. Moll, "The da Vinci telerobotic surgical system: The virtual operative field and telepresence surgery," *Surgical Clinics of North America*, vol. 83, no. 6, pp. 1293–1304, 2003.

[24] T. Aykut, J. Xu, and E. Steinbach, "Realtime 3D 360-Degree Telepresence With Deep-Learning-Based Head-Motion Prediction," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 231–244, 2019.

[25] K. Theofilis, J. Orlosky, Y. Nagai, and K. Kiyokawa, "Panoramic view reconstruction for stereoscopic teleoperation of a humanoid robot," *IEEE-RAS International Conference on Humanoid Robots*, pp. 242–248, 2016.

[26] S. Arevalo and F. Rucker, "Assisting manipulation and grasping in robot teleoperation with augmented reality visual cues," in *Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery, may 2021.

[27] E. Rosen, N. Kumar, D. Whitney, G. Konidaris, D. Ullman, and S. Tellex, "Bridging the Semantic Gap for Robots: Action-Oriented Semantic Maps via Mixed Reality,"

[28] C. P. Quintero, S. Li, M. K. Pan, W. P. Chan, H. F. Machiel Van Der Loos, and E. Croft, "Robot Programming Through Augmented Trajectories in Augmented Reality," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1838–1844, 2018.

[29] J. Guhl, J. Hügle, and J. Krüger, "Enabling human-robot-interaction via virtual and augmented reality in distributed control systems," in *Procedia CIRP*, vol. 76, pp. 167–170, Elsevier B.V., 2018.

[30] K. R. Guerin, S. D. Riedel, J. Bohren, and G. D. Hager, "Adjutant: A framework for flexible human-machine collaborative systems," *IEEE International Conference on Intelligent Robots and Systems*, no. Iros, pp. 1392–1399, 2014.

[31] T. Dardona, S. Eslamian, L. A. Reisner, and A. Pandya, "Remote presence: Development and usability evaluation of a head-mounted display for camera control on the da Vinci Surgical System," *Robotics*, vol. 8, no. 2, 2019.

[32] S. Kohlbecher, K. Bartl, S. Bardins, and E. Schneider, "Low-latency combined eye and head tracking system for teleoperating a robotic head in real-time," *Eye Tracking Research and Applications Symposium (ETRA)*, vol. 1, no. 212, pp. 117–120, 2010.

[33] N. Tran, J. Rands, and T. Williams, "A Hands-Free Virtual-Reality Teleoperation Interface for Wizard-of-Oz Control," *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI*, no. March, 2018.

[34] H. Martins, I. Oakley, and R. Ventura, "Design and evaluation of a head-mounted display for immersive 3D teleoperation of field robots," *Robotica*, vol. 33, pp. 2166–2185, dec 2015.

[35] G. Doisy, A. Ronen, and Y. Edan, "Comparison of three different techniques for camera and motion control of a teleoperated robot," *Applied Ergonomics*, vol. 58, pp. 527–534, jan 2017.

[36] B. MacKey, P. Bremner, and M. Giuliani, "Usability of an Immersive Control System for a Humanoid Robot Surrogate," *RO-MAN 2022 - 31st IEEE International Conference on Robot and Human Interactive Communication: Social, Asocial, and Antisocial Robots*, pp. 678–685, 2022.

[37] Y. Oh, R. Parasuraman, T. Mcgraw, and B.-C. Min, "360 VR Based Robot Teleoperation Interface for Virtual Tour," *International Workshop on Virtual, Augmented and Mixed Reality for Human-Robot Interaction*, no. March, 2018.

[38] T. Kot and P. Novák, "Utilization of the oculus rift HMD in mobile robot teleoperation," *Applied Mechanics and Materials*, vol. 555, no. November 2018, pp. 199–208, 2014.

[39] J. Soares, A. Vale, and R. Ventura, "A Multi-purpose Rescue Vehicle and a human-robot interface architecture for remote assistance in ITER," *Fusion Engineering and Design*, vol. 98-99, pp. 1656–1659, oct 2015.

[40] T. Kot and P. Novák, "Application of virtual reality in teleoperation of the military mobile robotic system TAROS," *International Journal of Advanced Robotic Systems*, vol. 15, jan 2018.

[41] B. Bejczy, R. Bozyil, E. Vaiekauskas, S. B. K. Petersen, S. Bogh, S. S. Hjorth, and E. B. Hansen, "Mixed reality interface for improving mobile manipulator teleoperation in contamination critical applications," in *Procedia Manufacturing*, vol. 51, pp. 620–626, Elsevier B.V., 2020.

[42] J. I. Lipton, A. J. Fay, and D. Rus, "Baxter's Homunculus: Virtual Reality Spaces for Teleoperation in Manufacturing," *IEEE Robotics and Automation Letters*, vol. 3, pp. 179–186, jan 2018.

[43] A. W. Yew, S. K. Ong, and A. Y. Nee, "Immersive Augmented Reality Environment for the Teleoperation of Maintenance Robots," in *Procedia CIRP*, vol. 61, pp. 305–310, Elsevier B.V., 2017.

[44] D. Krupke, F. Steinicke, P. Lubos, Y. Jonetzko, M. Gorner, and J. Zhang, "Comparison of Multimodal Heading and Pointing Gestures for Co-Located Mixed Reality Human-Robot Interaction," *IEEE International Conference on Intelligent Robots and Systems*, no. i, pp. 5003–5009, 2018.

[45] J. D. Hernandez, S. Sobti, A. Sciola, M. Moll, and L. E. Kavraki, "Increasing robot autonomy via motion planning and an augmented reality interface," *IEEE Robotics and Automation Letters*, vol. 5, pp. 1017–1023, apr 2020.

[46] E. Rosen, D. Whitney, E. Phillips, G. Chien, J. Tompkin, G. Konidaris, and S. Tellex, "Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays," *International Journal of Robotics Research*, 2019.

[47] M. E. Walker, H. Hedayati, and D. Szafir, "Robot Teleoperation with Augmented Reality Virtual Surrogates," *ACM/IEEE International Conference on Human-Robot Interaction*, vol. 2019-March, pp. 202–210, 2019.

[48] M. Ostanin, S. Mikhel, A. Evlampiev, V. Skvortsova, and A. Klimchik, "Human-robot interaction for robotic manipulator programming in Mixed Reality," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2805–2811, IEEE, may 2020.

[49] J. O. Neill, S. Ourselin, T. Vercauteren, L. Cruz, and C. Bergeles, "Virtual Reality to Enhance Surgical Robot Development and Control,"

[50] V. Vunder, R. Valuer, C. McMahon, K. Kruusamae, and M. Pryor, "Improved situational awareness in ROS using panospheric vision and virtual reality," *Proceedings - 2018 11th International Conference on Human System Interaction, HSI 2018*, pp. 471–477, 2018.

[51] S. I. Ktena, W. Abbott, and A. A. Faisal, "A virtual reality platform for safe evaluation and training of natural gaze-based wheelchair driving," in *International IEEE/EMBS Conference on Neural Engineering, NER*, vol. 2015-July, pp. 236–239, IEEE Computer Society, jul 2015.

[52] B. Omarali, T. Taunyazov, A. Bukeyev, and A. Shintemirov, "Real-time predictive control of an UR5 robotic arm through human upper limb motion tracking," *ACM/IEEE International Conference on Human-Robot Interaction*, no. July 2019, p. 414, 2017.

[53] R. Codd-Downey, P. M. Forooshani, A. Speers, H. Wang, and M. Jenkin, "From ROS to unity: Leveraging robot and virtual environment middleware for immersive teleoperation," *2014 IEEE International Conference on Information and Automation, ICIA 2014*, no. July, pp. 932–936, 2014.

[54] M. Rubagotti, T. Taunyazov, B. Omarali, and A. Shintemirov, "Semi-Autonomous Robot Teleoperation With Obstacle Avoidance via Model Predictive Control," *IEEE Robotics and Automation Letters*, vol. 4, pp. 2746–2753, jul 2019.

[55] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, "Rosbridge: ROS for Non-ROS Users," in *Robotics Research*, vol. 100, pp. 493–504, 2017.

[56] T. Rodehutskors, M. Schwarz, and S. Behnke, "Intuitive bimanual telemanipulation under communication restrictions by immersive 3D visualization and motion tracking," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Decem, pp. 276–283, 2015.

[57] S. Gray, R. Chevalier, D. Kotfis, B. Caimano, K. Chaney, A. Rubin, K. Fregene, and T. Danko, "An architecture for human-guided autonomy: Team TROOPER at the DARPA robotics challenge finals," *Springer Tracts in Advanced Robotics*, vol. 121, no. 0, pp. 549–582, 2018.

[58] Y. Mae, T. Inoue, K. Kamiyama, M. Kojima, M. Horade, and T. Arai, "Direct teleoperation system of multi-limbed robot for moving on complicated environments," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1171–1174, IEEE, dec 2017.

[59] B. Vagvolgyi, W. Niu, Z. Chen, P. Wilkening, and P. Kazanzides, "Augmented virtuality for model-based teleoperation," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 3826–3833, 2017.

[60] Y. Horikawa, A. Egashira, K. Nakashima, A. Kawamura, and R. Kurazume, "Previewed reality: Near-future perception system," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 370–375, IEEE, sep 2017.

[61] S. Feichter and H. Hlavacs, "Planar Simplification of Indoor Point-Cloud Environments," in *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 274–281, IEEE, dec 2018.

[62] T. Zhou, Q. Zhu, and J. Du, "Intuitive robot teleoperation for civil engineering operations with virtual reality and deep learning scene reconstruction," *Advanced Engineering Informatics*, vol. 46, no. July, p. 101170, 2020.

[63] R. Tredinnick, M. Broecker, and K. Ponto, "Progressive Feedback Point Cloud Rendering for Virtual Reality Display," *2016 IEEE Virtual Reality (VR)*, pp. 301–302.

[64] S. Kim, Y. Kim, J. Ha, and S. Jo, "Mapping System with Virtual Reality for Mobile Robot Teleoperation," *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, no. Iccas, p. 1541, 2018.

[65] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling," *CVGIP: Graphical Models and Image Processing*, vol. 53, pp. 157–185, oct 2016.

[66] J. Thomason, P. Ratsamee, J. Orlosky, K. Kiyokawa, T. Mashita, Y. Uranishi, and H. Takemura, "A Comparison of Adaptive View Techniques for Exploratory 3D Drone Teleoperation," *ACM Transactions on Interactive Intelligent Systems*, vol. 9, no. 2-3, pp. 1–19, 2019.

[67] S. Luo, W. Mou, K. Althoefer, and H. Liu, "Iterative Closest Labeled Point for tactile object shape recognition," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 3137–3142, 2016.

[68] G. Izatt, G. Mirano, E. Adelson, and R. Tedrake, "Tracking objects with point clouds from vision and touch," in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4000–4007, Institute of Electrical and Electronics Engineers Inc., jul 2017.

[69] Y. He and S. Chen, "Recent Advances in 3D Data Acquisition and Processing by Time-of-Flight Camera," *IEEE Access*, vol. 7, pp. 12495–12510, 2019.

[70] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1–4, 2011.

[71] D. Wei, B. Huang, and Q. Li, "Multi-view merging for robot teleoperation with virtual reality," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8537–8544, 2021.

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

[73] F. Okura, Y. Ueda, T. Sato, and N. Yokoya, "[Paper] Free-viewpoint Mobile Robot Teleoperation Interface Using View-dependent Geometry and Texture," *ITE Transactions on Media Technology and Applications*, vol. 2, no. 1, pp. 82–93, 2014.

[74] A. Al-Nuaimi, E. Steinbach, W. B. Lopes, and C. G. Lopes, "6DOF point cloud alignment using geometric algebra-based adaptive filtering," *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*, pp. 1–9, 2016.

[75] S. Beck and B. Froehlich, "Sweeping-based volumetric calibration and registration of multiple RGBD-sensors for 3D capturing systems," in *2017 IEEE Virtual Reality (VR)*, pp. 167–176, IEEE, 2017.

[76] D. Nicolis, M. Palumbo, A. M. Zanchettin, and P. Rocco, "Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots," *IEEE Robotics and Automation Letters*, vol. 3, pp. 796–803, apr 2018.

[77] D. Rakita, B. Mutlu, and M. Gleicher, "Remote Telemanipulation with Adapting Viewpoints in Visually Complex Environments," in *Robotics: Science and Systems XV*, Robotics: Science and Systems Foundation, jun 2019.

[78] M. Draelos, B. Keller, C. Toth, A. Kuo, K. Hauser, and J. Izatt, "Teleoperating robots from arbitrary viewpoints in surgical contexts," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2549–2555, IEEE, sep 2017.

[79] Y. H. Su, K. Huang, and B. Hannaford, "Multicamera 3D Reconstruction of Dynamic Surgical Cavities: Autonomous Optimal Camera Viewpoint Adjustment," in *2020 International Symposium on Medical Robotics, ISMR 2020*, pp. 103–110, Institute of Electrical and Electronics Engineers Inc., nov 2020.

[80] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3828–3836, 2015.

[81] J. Xue, H. Zhang, K. Dana, and K. Nishino, "Differential angular imaging for material recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 764–773, 2017.

[82] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Material recognition in the wild with the materials in context database," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3479–3487, 2015.

[83] I. Vitanov, I. Farkhatdinov, B. Denoun, F. Palermo, A. Otaran, J. Brown, B. Omarali, T. Abrar, M. Hansard, C. Oh, *et al.*, "A suite of robotic solutions for nuclear waste decommissioning," *Robotics*, vol. 10, no. 4, p. 112, 2021.

[84] F. De Boissieu, C. Godin, B. Guilhamat, D. David, C. Serviere, and D. Baudois, "Tactile texture recognition with a 3-axial force mems integrated artificial finger.," in *Robotics: Science and Systems*, pp. 49–56, Seattle, WA, 2009.

[85] A. Drimus, M. B. Petersen, and A. Bilberg, "Object texture recognition by dynamic tactile sensing using active exploration," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 277–283, IEEE, 2012.

[86] T. P. Tomo, S. Somlor, A. Schmitz, L. Jamone, W. Huang, H. Kristanto, and S. Sugano, "Design and characterization of a three-axis hall effect-based soft skin sensor," *Sensors*, vol. 16, no. 4, p. 491, 2016.

[87] J. Konstantinova, G. Cotugno, A. Stilli, Y. Noh, and K. Althoefer, "Object classification using hybrid fiber optical force/proximity sensor," in *2017 IEEE SENSORS*, (Glasgow, UK), pp. 1–3, IEEE, 2017.

[88] F. Palermo, J. Konstantinova, K. Althoefer, S. Poslad, and I. Farkhatdinov, "Implementing tactile and proximity sensing for crack detection," in *IEEE International Conference on Robotics and Automation*, 2020.

[89] F. Palermo, J. Konstantinova, K. Althoefer, S. Poslad, and I. Farkhatdinov, "Automatic fracture characterization using tactile and proximity optical sensing," *Frontiers in Robotics and AI*, vol. 7, 2020.

[90] F. Palermo, L. Rincon-Ardila, C. Oh, K. Althoefer, S. Poslad, G. Venture, and I. Farkhatdinov, "Multi-modal robotic visual-tactile localisation and detection of surface cracks," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pp. 1806–1811, IEEE, 2021.

[91] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[92] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[93] S. S. Baishya and B. Bäuml, "Robust material classification with a tactile skin using deep learning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8–15, IEEE, 2016.

[94] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, "Deep learning for tactile understanding from visual and haptic data," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 536–543, IEEE, 2016.

[95] M. Alameh, A. Ibrahim, M. Valle, and G. Moser, "Dcnn for tactile sensory data classification based on transfer learning," in *2019 15th Conference on Ph. D Research in Microelectronics and Electronics (PRIME)*, pp. 237–240, IEEE, 2019.

[96] H. Buhler, S. Misztal, and J. Schild, "Reducing vr sickness through peripheral visual effects," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 517–9, 2018.

[97] E. R. Hoeg, K. V. Ruder, N. C. Nilsson, R. Nordahl, and S. Serafin, "An exploration of input conditions for virtual teleportation," in *2017 IEEE Virtual Reality (VR)*, pp. 341–342, 2017.

[98] C. Park and K. Jang, "Investigation of visual self-representation for a walking-in-place navigation system in virtual reality," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1114–1115, 2019.

[99] D. Rakita, "Methods for Effective Mimicry-based Teleoperation of Robot Arms," *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17*, pp. 371–372, 2017.

[100] J. Kofman, X. Wu, T. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 5, pp. 1206–1219, 2005.

[101] H. Reddivari, C. Yang, Z. Ju, P. Liang, Z. Li, and B. Xu, "Teleoperation control of Baxter robot using body motion tracking," in *2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, vol. 30, pp. 1–6, IEEE, sep 2014.

[102] L. Peppoloni, F. Brizzi, C. A. Avizzano, and E. Ruffaldi, "Immersive ROS-integrated framework for robot teleoperation," *2015 IEEE Symposium on 3D User Interfaces, 3DUI 2015 - Proceedings*, pp. 177–178, 2015.

[103] D. Sun, A. Kiselev, Q. Liao, T. Stoyanov, and A. Loutfi, "A New Mixed-Reality-Based Teleoperation System for Telepresence and Maneuverability Enhancement," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 1, pp. 55–67, 2020.

[104] Y.-h. Su, I. Huang, K. Huang, and B. Hannaford, "Comparison of 3D Surgical Tool Segmentation Procedures with Robot Kinematics Prior," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4411–4418, IEEE, oct 2018.

[105] M. Di Castro, D. B. Mulero, M. Ferre, and A. Masi, "A Real-Time Reconfigurable Collision Avoidance System for Robot Manipulation," *Proceedings of the 3rd International Conference on Mechatronics and Robotics Engineering - ICMRE 2017*, pp. 6–10, 2017.

[106] A. Leeper, K. Hsiao, M. Ciocarlie, I. Sucan, and K. Salisbury, "Methods for Collision-Free Arm Teleoperation in Clutter Using Constraints from 3D Sensor Data," *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 520–527, 2013.

[107] C. Yang, X. Wang, L. Cheng, and H. Ma, "Neural-Learning-Based Telerobot Control with Guaranteed Performance," *IEEE Transactions on Cybernetics*, vol. 47, pp. 3148–3159, oct 2017.

[108] D. Rakita, B. Mutlu, and M. Gleicher, "RelaxedIK: Real-time Synthesis of Accurate and Feasible Robot Arm Motion," in *Robotics: Science and Systems XIV*, pp. 1–9, Robotics: Science and Systems Foundation, jun 2018.

[109] S. Gaurav, Z. Al-Qurashi, A. Barapatre, G. Maratos, T. Sarma, and B. D. Ziebart, "Deep Correspondence Learning for Effective Robotic Teleoperation using Virtual Reality," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pp. 477–483, IEEE, oct 2019.

[110] Z. Al-Qurashi and B. D. Ziebart, "Recurrent Neural Networks for Hierarchically Mapping Human-Robot Poses," in *Proceedings - 4th IEEE International Conference on Robotic Computing, IRC 2020*, pp. 63–70, Institute of Electrical and Electronics Engineers Inc., nov 2020.

[111] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep Learning for 3D Point Clouds: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 4338–4364, dec 2021.

[112] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "SEGCloud: Semantic Segmentation of 3D Point Clouds," in *2017 International Conference on 3D Vision (3DV)*, pp. 537–547, IEEE, oct 2017.

[113] D. Bobkov, S. Chen, M. Kiechle, S. Hilsenbeck, and E. Steinbach, "Noise-resistant Unsupervised Object Segmentation in Multi-view Indoor Point Clouds," *VISIGRAPP 2017 - Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 5, no. Visigrapp, pp. 149–156, 2017.

[114] D. Bobkov, S. Chen, R. Jian, M. Z. Iqbal, and E. Steinbach, "Noise-Resistant Deep Learning for Object Classification in Three-Dimensional Point Clouds Using a Point Pair Descriptor," *IEEE Robotics and Automation Letters*, vol. 3, pp. 865–872, apr 2018.

[115] Z. Liang, M. Yang, L. Deng, C. Wang, and B. Wang, "Hierarchical Depthwise Graph Convolutional Neural Network for 3D Semantic Segmentation of Point Clouds," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8152–8158, IEEE, may 2019.

[116] J. Hou, A. Dai, and M. Niebner, "3D-SIS: 3D semantic instance segmentation of RGB-D scans," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 4416–4425, 2019.

[117] Y. Ishikawa, "Semantic Segmentation of 3D Point Cloud to Virtually Manipulate Real Living Space," *2019 12th Asia Pacific Workshop on Mixed and Augmented Reality (APMAR)*, pp. 1–7.

[118] Q. H. Pham, B. S. Hua, D. T. Nguyen, and S. K. Yeung, "Real-time progressive 3D semantic segmentation for indoor scenes," in *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019*, pp. 1089–1098, Institute of Electrical and Electronics Engineers Inc., mar 2019.

[119] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting Unknown 3D Objects from Real Depth Images using Mask R-CNN Trained on Synthetic Data," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7283–7290, IEEE, may 2019.

[120] X. Wang, C. Yang, and H. Ma, "Automatic obstacle avoidance using redundancy for shared controlled telerobot manipulator," *2015 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2015*, pp. 1338–1343, 2015.

[121] B. Rydén, Fredrik and Chizeck, Howard and nia kosari, Sina and King, H. Hawkeye and Hannaford, "Using Kinect TM and a Haptic Interface for Implementation of Real-Time Virtual Fixtures," 2011.

[122] A. Leeper, S. Chan, K. Hsiao, M. Ciocarlie, and K. Salisbury, "Constraint-based haptic rendering of point data for teleoperated robot grasping," *Haptics Symposium 2012, HAPTICS 2012 - Proceedings*, pp. 377–383, 2012.

[123] A. Leeper, S. Chan, and K. Salisbury, "Point Clouds Can Be Represented as Implicit Surfaces for Constraint-Based Haptic Rendering," *2012 IEEE International Conference on Robotics and Automation*, pp. 5000–5005, 2012.

[124] X. Xu, B. Cizmeci, and E. Steinbach, "Point-cloud-based model-mediated teleoperation," *HAVE 2013 - 2013 IEEE International Symposium on Haptic Audio-Visual Environments and Games, Proceedings*, pp. 69–74, 2013.

[125] X. Xu, B. Cizmeci, A. Al-Nuaimi, and E. Steinbach, "Point cloud-based model-mediated teleoperation with dynamic and perception-based model updating," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 11, pp. 2558–2569, 2014.

[126] X. Xu, B. Cizmeci, C. Schuwerk, and E. Steinbach, "Model-Mediated Teleoperation: Toward Stable and Transparent Teleoperation Systems," *IEEE Access*, vol. 4, pp. 425–449, 2016.

[127] D. Ni, A. Song, X. Xu, H. Li, C. Zhu, and H. Zeng, "3D-point-cloud registration and real-world dynamic modelling-based virtual environment building method for teleoperation," *Robotica*, vol. 35, pp. 1958–1974, oct 2017.

[128] D. Valenzuela-Urrutia, R. Muñoz-Riffo, and J. Ruiz-del Solar, "Virtual Reality-Based Time-Delayed Haptic Teleoperation Using Point Cloud Data," *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2019.

[129] S. Kim and J. Park, "A Unified Virtual Fixture Model for Haptic Telepresence Systems based on Streaming Point Cloud Data and Implicit Surfaces," *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, no. Iccas, pp. 881–885, 2016.

[130] D. Lee and Y. S. Park, "Implementation of Augmented Teleoperation System Based on Robot Operating System (ROS)," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2018-Janua, pp. 5497–5502, IEEE, oct 2018.

[131] M. Selvaggio, G. Notomista, F. Chen, B. Gao, F. Trapani, and D. Caldwell, "Enhancing bilateral teleoperation using camera-based online virtual fixtures generation," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 1483–1488, 2016.

[132] C. P. Quintero, M. Dehghan, O. Ramirez, M. H. Ang, and M. Jagersand, "Flexible virtual fixture interface for path specification in tele-manipulation," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5363–5368, 2017.

[133] T. Stoyanov, R. Krug, A. Kiselev, D. Sun, and A. Loutfi, "Assisted Telemanipulation: A Stack-Of-Tasks Approach to Remote Manipulator Control," *IEEE International Conference on Intelligent Robots and Systems*, no. January, pp. 6640–6645, 2018.

[134] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization for teleoperation and teaming," *The International Journal of Robotics Research*, vol. 37, pp. 717–742, jun 2018.

[135] D. Ni, A. Y. Nee, S. K. Ong, H. Li, C. Zhu, and A. Song, "Point cloud augmented virtual reality environment with haptic constraints for teleoperation," *Transactions of the Institute of Measurement and Control*, pp. 1–14, 2018.

[136] V. Pruks, K.-H. Lee, and J.-H. Ryu, "Shared Teleoperation for Nuclear Plant Robotics Using Interactive Virtual Guidance Generation and Shared Autonomy Approaches," in *2018 15th International Conference on Ubiquitous Robots (UR)*, pp. 91–95, IEEE, jun 2018.

[137] V. Pruks and J. H. Ryu, "Interactive Virtual Fixture Generation for Shared Teleoperation in Unstructured Environments," *Lecture Notes in Electrical Engineering*, vol. 535, pp. 88–91, 2019.

[138] V. Pruks and J.-H. Ryu, "A Framework for Interactive Virtual Fixture Generation for Shared Teleoperation in Unstructured Environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10234–10241, IEEE, may 2020.

[139] M. R. Wrock, S. Nokleby, M. R. Wrock, and S. B. Nokleby, "Haptic Teleoperation of a Manipulator using Virtual Fixtures and Hybrid Position-Velocity Control," tech. rep.

[140] Z. Wang, I. Reed, and A. M. Fey, "Toward Intuitive Teleoperation in Surgery: Human-Centric Evaluation of Teleoperation Algorithms for Robotic Needle Steering," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, IEEE, may 2018.

[141] B. Xi, S. Wang, X. Ye, Y. Cai, T. Lu, and R. Wang, "A robotic shared control teleoperation method based on learning from demonstrations," *International Journal of Advanced Robotic Systems*, vol. 16, no. 4, p. 172988141985742, 2019.

[142] S. Nikolaidis, Y. X. Zhu, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in shared autonomy," pp. 294–302, 2017.

[143] U. Acharya, S. Kunde, L. Hall, B. A. Duncan, and J. M. Bradley, "Inference of User Qualities in Shared Control," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 588–595, IEEE, may 2018.

[144] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, p. 1, 2012.

[145] J. Lambrecht and J. Kruger, "Spatial programming for industrial robots based on gestures and Augmented Reality," *IEEE International Conference on Intelligent Robots and Systems*, pp. 466–472, 2012.

[146] A. Gaschler, M. Springer, M. Rickert, and A. Knoll, "Intuitive robot Tasks with augmented reality and virtual obstacles," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 6026–6031, 2014.

[147] J. Vaughan, S. Kratz, and D. Kimber, "Look where you're going: Visual interfaces for robot teleoperation," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 273–280, IEEE, aug 2016.

[148] A. Leeper, K. Hsiao, E. Chu, and J. K. Salisbury, *Using Near-Field Stereo Vision for Robotic Grasping in Cluttered Environments*, vol. 79 of *Springer Tracts in Advanced Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[149] D. Kent, C. Saldanha, and S. Chernova, "A Comparison of Remote Robot Teleoperation Interfaces for General Object Manipulation," in *ACM/IEEE International Conference on Human-Robot Interaction*, vol. Part F1271, pp. 371–379, IEEE Computer Society, mar 2017.

[150] A. Makhal, F. Thomas, and A. P. Gracia, "Grasping Unknown Objects in Clutter by Superquadric Representation," in *2018 Second IEEE International Conference on Robotic Computing (IRC)*, vol. 2018-Janua, pp. 292–299, IEEE, jan 2018.

[151] D. Kent, C. Saldanha, and S. Chernova, "Leveraging depth data in remote robot teleoperation interfaces for general object manipulation," *International Journal of Robotics Research*, vol. 39, pp. 39–53, jan 2020.

[152] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *International Journal of Robotics Research*, vol. 39, pp. 183–201, mar 2020.

[153] D. L. Chow, P. Xu, E. Tuna, S. Huang, M. C. Cavusoglu, and W. Newman, "Supervisory control of a DaVinci surgical robot," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 5043–5049, 2017.

[154] R. Hetrick, N. Amerson, B. Kim, E. Rosen, E. J. de Visser, and E. Phillips, "Comparing Virtual Reality Interfaces for the Teleoperation of Robots," in *2020 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 1–7, IEEE, apr 2020.

[155] G. Baker, T. Bridgwater, P. Bremner, and M. Giuliani, "Towards an immersive user interface for waypoint navigation of a mobile robot," 2020.

[156] I. Havoutis and S. Calinon, "Supervisory teleoperation with online learning and optimal control," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1534–1540, 2017.

[157] M. Rigter, B. Lacerda, and N. Hawes, "A Framework for Learning from Demonstration with Minimal Human Effort," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2023–2030, 2020.

[158] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, IEEE, may 2018.

[159] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 185–207, 2013.

[160] R. Zhang, A. Saran, B. Liu, Y. Zhu, S. Guo, S. Niekum, D. Ballard, and M. Hayhoe, "Human gaze assisted artificial intelligence: A review," in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2021-Janua, pp. 4951–4958, International Joint Conferences on Artificial Intelligence, 2020.

[161] R. M. Hecht, A. B. Hillel, A. Telpaz, O. Tsimhoni, and N. Tishby, "Information Constrained Control Analysis of Eye Gaze Distribution under Workload," *IEEE Transactions on Human-Machine Systems*, vol. 49, pp. 474–484, dec 2019.

[162] C. K. Yin Li, Xiaodi Hou, "The Secrets of Salient Object Segmentation Supplementary Materials," *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),*, pp. 4321–4328, 2014.

[163] Z. Bylinskii, P. Isola, C. Bainbridge, A. Torralba, and A. Oliva, "Intrinsic and extrinsic effects on image memorability," *Vision Research*, vol. 116, pp. 165–178, 2015.

[164] C. Ozcinar and A. Smolic, "Visual Attention in Omnidirectional Video for Virtual Reality Applications," in *2018 10th International Conference on Quality of Multimedia Experience, QoMEX 2018*, Institute of Electrical and Electronics Engineers Inc., sep 2018.

[165] X. Zhang, X. Zhu, X.-y. Zhang, N. Zhang, P. Li, and L. Wang, "SegGAN : Semantic Segmentation with Generative Adversarial Network," *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, pp. 1–5, 2018.

[166] Z. Zhang, Y. Xu, J. Yu, and S. Gao, "Saliency detection in 360° Videos," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11211 LNCS, pp. 504–520, 2018.

[167] Y. Xu, C. Yang, J. Zhong, H. Ma, L. Zhao, and M. Wang, "Robot teaching by teleoperation based on visual interaction and neural network learning," *Proceedings of 2017 9th International Conference On Modelling, Identification and Control, ICMIC 2017*, vol. 2018-March, pp. 1068–1073, 2018.

[168] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, "Gaze Prediction in Dynamic 360," *Cvpr*, pp. 5333–5342, 2018.

[169] A. Palazzi, D. Abati, S. Calderara, F. Solera, and R. Cucchiara, "Predicting the Driver's Focus of Attention: The DR(eye)VE Project," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1720–1733, 2019.

[170] G. Ziv, "Gaze Behavior and Visual Attention: A Review of Eye Tracking Studies in Aviation," *The International Journal of Aviation Psychology*, vol. 26, pp. 75–104, oct 2016.

[171] R. Zhang, C. Walshe, Z. Liu, L. Guan, K. S. Muller, J. A. Whritner, L. Zhang, M. M. Hayhoe, and D. H. Ballard, "Atari-HEAD: Atari human eye-tracking and demonstration dataset," *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pp. 6811–6820, 2020.

[172] K. Ruhland, C. E. Peters, S. Andrist, J. B. Badler, N. I. Badler, M. Gleicher, B. Mutlu, and R. McDonnell, "A Review of Eye Gaze in Virtual Agents, Social Robotics and HCI: Behaviour Generation, User Interaction and Perception," *Computer Graphics Forum*, vol. 34, no. 6, pp. 299–326, 2015.

[173] M. Giuliani, D. Szczęśniak-Stańczyk, N. Mirnig, G. Stollnberger, M. Szyszko, B. Stańczyk, and M. Tscheligi, "User-centred design and evaluation of a tele-operated echocardiography robot," *Health and Technology*, vol. 10, no. 3, pp. 649–665, 2020.

[174] E. B. Lavoie, A. M. Valevicius, Q. A. Boser, O. Kovic, A. H. Vette, P. M. Pilarski, J. S. Hebert, and C. S. Chapman, "Using synchronized eye and motion tracking to determine high-precision eye-movement patterns during objectinteraction tasks," *Journal of Vision*, vol. 18, no. 6, pp. 1–20, 2018.

[175] Z. Hu, S. Li, and M. Gai, "Temporal continuity of visual attention for future gaze prediction in immersive virtual reality," *Virtual Reality and Intelligent Hardware*, vol. 2, pp. 142–152, apr 2020.

[176] F. Berton, A.-H. Olivier, J. Bruneau, L. Hoyet, J. Pettré, eì ene Olivier, and J. Pettre, "Studying Gaze Behaviour During Collision Avoidance With a Virtual Walker: Influence of the Virtual Reality Setup," pp. 717–725, 2019.

[177] J. Pettersson and P. Falkman, "Human movement direction classification using virtual reality and eye tracking," in *Procedia Manufacturing*, vol. 51, pp. 95–102, Elsevier B.V., 2020.

[178] J. Pettersson and P. Falkman, "Human Movement Direction Prediction using Virtual Reality and Eye Tracking," *Proceedings of the IEEE International Conference on Industrial Technology*, vol. 2021-March, pp. 889–894, 2021.

[179] B. David-John, C. Peacock, T. Zhang, T. S. Murdison, H. Benko, and T. R. Jonker, "Towards gaze-based prediction of the intent to interact in virtual reality," in *Eye Tracking Research and Applications Symposium (ETRA)*, vol. PartF16925, Association for Computing Machinery, may 2021.

[180] R. Zhang, S. Zhang, M. H. Tong, Y. Cui, C. A. Rothkopf, D. H. Ballard, and M. M. Hayhoe, "Modeling sensory-motor decisions in natural behavior," *PLoS Computational Biology*, vol. 14, oct 2018.

[181] M. Yu, X. Wang, Y. Lin, and X. Bai, "Gaze tracking system for teleoperation," in *26th Chinese Control and Decision Conference, CCDC 2014*, pp. 4617–4622, IEEE Computer Society, 2014.

[182] L. Yuan, C. Reardon, G. Warnell, and G. Loianno, "Human gaze-driven spatial tasking of an autonomous MAV," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1343–1350, 2019.

[183] J. Guo, Y. Liu, Q. Qiu, J. Huang, C. Liu, Z. Cao, and Y. Chen, "A Novel Robotic Guidance System with Eye-Gaze Tracking Control for Needle-Based Interventions," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 1, pp. 179–188, 2021.

[184] S. Miura, R. Ohta, Y. Cao, K. Kawamura, Y. Kobayashi, and M. G. Fujie, "Using Operator Gaze Tracking to Design Wrist Mechanism for Surgical Robots," *IEEE Transactions on Human-Machine Systems*, vol. 51, pp. 376–383, aug 2021.

[185] C. Antonya, F. G. Bărbuceanu, and Z. Rusák, "Path generation in virtual reality environment based on gaze analysis," *IEEE AFRICON Conference*, no. September, pp. 13–15, 2011.

[186] L. Scalera, S. Seriani, P. Gallina, M. Lentini, and A. Gasparetto, "Human–Robot Interaction through Eye Tracking for Artistic Drawing," *Robotics*, vol. 10, p. 54, mar 2021.

[187] L. Scalera, S. Seriani, P. Gallina, M. Lentini, and A. Gasparetto, "Human–robot interaction through eye tracking for artistic drawing," *Robotics*, vol. 10, jun 2021.

[188] H. O. Latif, N. Sherkat, and A. Lotfi, "Teleoperation through eye gaze (TeleGaze): A multimodal approach," *2009 IEEE International Conference on Robotics and Biomimetics, ROBIO 2009*, no. September 2016, pp. 711–716, 2009.

[189] D. Gego, C. Carreto, and L. Figueiredo, "Teleoperation of a mobile robot based on eye-gaze tracking," *Iberian Conference on Information Systems and Technologies, CISTI*, 2017.

[190] M. Minamoto, Y. Suzuki, T. Kanno, and K. Kawashima, "Effect of robot operation by a camera with the eye tracking control," *2017 IEEE International Conference on Mechatronics and Automation, ICMA 2017*, pp. 1983–1988, 2017.

[191] A. Ezzat, A. Kogkas, J. Holt, R. Thakkar, A. Darzi, and G. Mylonas, "An eye-tracking based robotic scrub nurse: proof of concept," *Surgical Endoscopy*, vol. 35, no. 9, pp. 5381–5391, 2021.

[192] G. Zhang and J. P. Hansen, "Accessible Control of Telepresence Robots based on Eye Tracking," *Eye Tracking Research and Applications Symposium (ETRA)*, 2019.

[193] J. M. Araujo, G. Zhang, J. P. P. Hansen, and S. Puthusserypady, "Exploring Eye-Gaze Wheelchair Control," *Eye Tracking Research and Applications Symposium (ETRA)*, 2020.

[194] T. Piumsomboon, G. Lee, R. W. Lindeman, and M. Billinghurst, "Exploring natural eye-gaze-based interaction for immersive virtual reality," in *2017 IEEE Symposium on 3D User Interfaces, 3DUI 2017 - Proceedings*, pp. 36–39, Institute of Electrical and Electronics Engineers Inc., apr 2017.

[195] D. Zhu, T. Gedeon, and K. Taylor, "Exploring camera viewpoint control models for a multitasking setting in teleoperation," *Conference on Human Factors in Computing Systems - Proceedings*, pp. 53–62, 2011.

[196] D. Zhu, T. Gedeon, and K. Taylor, ""Moving to the centre": A gaze-driven remote camera control for teleoperation," *Interacting with Computers*, vol. 23, no. 1, pp. 85–95, 2011.

[197] A. Dünser, M. Lochner, U. Engelke, and D. R. Fernández, "Visual and manual control for human-robot teleoperation," *IEEE Computer Graphics and Applications*, vol. 35, pp. 22–32, may 2015.

[198] T. Carlson and Y. Demiris, "Using visual attention to evaluate collaborative control architectures for human robot interaction," *Adaptive and Emergent Behaviour and Complex*

Systems - Proceedings of the 23rd Convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour, AISB 2009, no. June 2014, pp. 38–43, 2009.

[199] T. Carlson and Y. Demiris, "Collaborative control for a robotic wheelchair: Evaluation of performance, attention, and workload," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 3, pp. 876–888, 2012.

[200] D. Novak and R. Riener, "Enhancing patient freedom in rehabilitation robotics using gaze-based intention detection," *IEEE International Conference on Rehabilitation Robotics*, pp. 1–6, 2013.

[201] P. M. Tostado, W. W. Abbott, and A. A. Faisal, "3D gaze cursor: Continuous calibration and end-point grasp control of robotic actuators," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3295–3300, IEEE, may 2016.

[202] A. Shafti, P. Orlov, and A. A. Faisal, "Gaze-based, Context-aware Robotic System for Assisted Reaching and Grasping," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 863–869, IEEE, may 2019.

[203] M. Y. Wang, A. A. Kogkas, A. Darzi, and G. P. Mylonas, "Free-View, 3D Gaze-Guided, Assistive Robotic System for Activities of Daily Living," *IEEE International Conference on Intelligent Robots and Systems*, pp. 2355–2361, 2018.

[204] Q. Zhu, J. Du, Y. Shi, and P. Wei, "Neurobehavioral assessment of force feedback simulation in industrial robotic teleoperation," *Automation in Construction*, vol. 126, no. March 2020, p. 103674, 2021.

[205] T. O. Zander, K. Shetty, R. Lorenz, D. R. Leff, L. R. Krol, A. W. Darzi, K. Gramann, and G. Z. Yang, "Automated Task Load Detection with Electroencephalography: Towards Passive Brain-Computer Interfacing in Robotic Surgery," *Journal of Medical Robotics Research*, vol. 2, no. 1, 2017.

[206] C. Berka, D. J. Levendowski, M. N. Lumicao, A. Yau, G. Davis, V. T. Zivkovic, R. E. Olmstead, P. D. Tremoulet, and P. L. Craven, "EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks.," *Aviation, space, and environmental medicine*, vol. 78, pp. B231–44, may 2007.

[207] Y. Guo, D. Freer, F. Deligianni, and G.-Z. Yang, "Eye-Tracking for Performance Evaluation and Workload Estimation in Space Telerobotic Training; Eye-Tracking for Performance Evaluation and Workload Estimation in Space Telerobotic Training," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 1, 2022.

[208] C. Wu, J. Cha, J. Sulek, T. Zhou, C. P. Sundaram, J. Wachs, and D. Yu, "Eye-Tracking Metrics Predict Perceived Workload in Robotic Surgical Skills Training," *Human Factors*, vol. 62, pp. 1365–1386, dec 2020.

[209] Y. Shi, N. Ruiz, R. Taib, E. Choi, and F. Chen, "Galvanic skin response (GSR) as an index of cognitive load," *Conference on Human Factors in Computing Systems - Proceedings*, pp. 2651–2656, 2007.

[210] R. Castaldo, P. Melillo, U. Bracale, M. Caserta, M. Triassi, and L. Pecchia, "Acute mental stress assessment via short term HRV analysis in healthy adults: A systematic review with meta-analysis," *Biomedical Signal Processing and Control*, vol. 18, pp. 370–377, 2015.

[211] M. Swangnetr and D. B. Kaber, "Emotional state classification in patient-robot interaction using wavelet analysis and statistics-based feature selection," *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 1, pp. 63–75, 2013.

[212] J. A. Healey and R. W. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, pp. 156–166, 2005.

[213] I. Nisky, M. H. Hsieh, and A. M. Okamura, "Uncontrolled Manifold Analysis of Arm Joint Angle Variability During Robotic Teleoperation and Freehand Movement of Surgeons and Novices," *IEEE Transactions on Biomedical Engineering*, vol. 61, pp. 2869–2881, dec 2014.

[214] D. Reinhardt, S. Haesler, J. Hurtienne, and C. Wienrich, "Entropy of Controller Movements Reflects Mental Workload in Virtual Reality," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 802–808, IEEE, mar 2019.

[215] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," pp. 139–183, 1988.

[216] D. Kent, C. Saldanha, and S. Chernova, "Leveraging depth data in remote robot teleoperation interfaces for general object manipulation," 2019.

[217] T. Tien, P. H. Pucher, M. H. Sodergren, K. Sriskandarajah, G. Z. Yang, and A. Darzi, "Eye tracking for skills assessment and training: A systematic review," 2014.

[218] S. Benedetto, M. Pedrotti, L. Minin, T. Baccino, A. Re, and R. Montanari, "Driver workload and eye blink duration," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 14, no. 3, pp. 199–208, 2011.

[219] J. Currie, R. R. Bond, P. McCullagh, P. Black, D. D. Finlay, and A. Peace, "Eye Tracking the Visual Attention of Nurses Interpreting Simulated Vital Signs Scenarios: Mining Metrics to Discriminate between Performance Level," *IEEE Transactions on Human-Machine Systems*, vol. 48, pp. 113–124, apr 2018.

[220] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, pp. 189–206, apr 2013.

[221] "Tilt brush."

[222] H. H. King, K. Tadano, R. Donlin, D. Friedman, M. J. H. Lum, V. Asch, C. Wang, K. Kawashima, and B. Hannaford, "Preliminary Protocol for Interoperable Telesurgery," *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1–6, 2009.

[223] N. Hogan, "Impedance Control: An Approach to Manipulation," in *1984 American Control Conference*, pp. 304–313, 1984.

[224] B. Denoun, B. Leon, C. Zito, R. Stolkin, L. Jamone, and M. Hansard, "Robust and fast generation of top and side grasps for unknown objects," 2019.

[225] "Tobii g2om."

[226] J. Konstantinova, A. Stilli, A. Faragasso, and K. Althoefer, "Fingertip proximity sensor with realtime visual-based calibration," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Daejeon, South Korea), pp. 170–175, IEEE, 2016.

[227] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[228] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24–31, 2016.

[229] S. Raghu, N. Sriraam, Y. Temel, S. V. Rao, and P. L. Kubben, "Eeg based multi-class seizure type classification using convolutional neural network and transfer learning," *Neural Networks*, vol. 124, pp. 202–212, 2020.

[230] Y.-H. Kwon, S.-B. Shin, and S.-D. Kim, "Electroencephalography based fusion two-dimensional (2d)-convolution neural networks (cnn) model for emotion recognition system," *Sensors*, vol. 18, no. 5, 2018.

[231] D. H. Gomez, J. R. Bagley, N. Bolter, M. Kern, and C. M. Lee, "Metabolic Cost and Exercise Intensity during Active Virtual Reality Gaming," *Games for Health Journal*, vol. 7, no. 5, pp. 310–316, 2018.

[232] W. Bu, G. Liu, and C. Liu, "Rate-position-point hybrid control mode for teleoperation with force feedback," *ICARM 2016 - 2016 International Conference on Advanced Robotics and Mechatronics*, pp. 420–425, 2016.

[233] I. Farkhatdinov, J.-H. Ryu, and J. Poduraev, "Control strategies and feedback information in mobile robot teleoperation," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 14681–14686, 2008.

[234] I. Farkhatdinov, J. H. Ryu, and J. Poduraev, "A user study of command strategies for mobile robot teleoperation," *Intelligent Service Robotics*, vol. 2, pp. 95–104, apr 2009.

[235] P. Chotiprayanakul and D. K. Liu, "Workspace mapping and force control for small haptic device based robot teleoperation," in *Int. Conf. on Information and Automation*, pp. 1613–1618, 2009.

[236] S. Stadler, H. Cornet, and F. Frenkler, "A study in virtual reality on (non-)gamers' attitudes and behaviors," *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings*, no. February, pp. 1169–1170, 2019.

[237] M. Sammut, M. Sammut, and P. Andrejevic, "The benefits of being a video gamer in laparoscopic surgery," *International Journal of Surgery*, vol. 45, pp. 42–46, 2017.

[238] I. Farkhatdinov, J.-H. Ryu, and J. Poduraev, "A user study of command strategies for mobile robot teleoperation," *Intelligent Service Robotics*, vol. 2, pp. 95–104, 2009.