Journal of Intelligent & Robotic Systems (2021) 101: 80
https://doi.org/10.1007/s10846-021-01311-7

REGULAR PAPER

The *Vicarios* Virtual Reality Interface for Remote Robotic Teleoperation

Teleporting for Intuitive Tele-manipulation

Abdeldjallil Naceri[1,2] · Dario Mazzanti[1] · Joao Bimbo[3] · Yonas T. Tefera[1] · Domenico Prattichizzo[4] · Darwin G. Caldwell[1] · Leonardo S. Mattos[1] · Nikhil Deshpande[1]

Received: 1 July 2020 / Accepted: 4 January 2021 / Published online: 3 April 2021

Abstract

Intuitive interaction is the cornerstone of accurate and effective performance in remote robotic teleoperation. It requires high-fidelity in control actions as well as perception (vision, haptic, and other sensory feedback) of the remote environment. This paper presents *Vicarios*, a Virtual Reality (VR) based interface with the aim of facilitating intuitive real-time remote teleoperation, while utilizing the inherent benefits of VR, including immersive visualization, freedom of user viewpoint selection, and fluidity of interaction through natural action interfaces. *Vicarios* aims to enhance the situational awareness, using the concept of *viewpoint-independent mapping* between the operator and the remote scene, thereby giving the operator better control in the perception-action loop. The article describes the overall system of *Vicarios*, with its software, hardware, and communication framework. A comparative user study quantifies the impact of the interface and its features, including immersion and instantaneous user viewpoint changes, termed "*teleporting*", on users' performance. The results show that users' performance with the VR-based interface was either similar to or better than the baseline condition of traditional stereo video feedback, approving the realistic nature of the *Vicarios* interface. Furthermore, including the teleporting feature in VR significantly improved participants' performance and their appreciation for it, which was evident in the post-questionnaire results. *Vicarios* capitalizes on the intuitiveness and flexibility of VR to improve accuracy in remote teleoperation.

Keywords Teleoperation · Teleporting · Telemanipulation · Immersion · Virtual reality · Motion mapping

1 Introduction

State-of-the-art systems in remote robotic teleoperation are able to execute complex, yet dexterous, tasks in demanding and/or hazardous environments, e.g., nuclear decommissioning, disaster response, search-and-rescue, surgery, etc. A recent example of this was the DARPA robotics challenge, where it was observed that a mixture of human-in-the-loop remote guidance and local robotic autonomy is the golden middle ground for telerobotics in adapting to unstructured, cluttered environments [23, 32, 39].

Although substantial research exists in the control of telerobotic systems [28], research into visual feedback of the remote environment that can significantly help and support effective teleoperation lags behind. Providing accurate and informative visual feedback is critical for successful teleoperation. However, the traditional user interfaces used for such guidance in standard teleoperation, relying mostly on stereo video feedback from the remote environment, suffer from limitations including accessibility, fixed or non-adaptable camera viewpoints, occluded views of the manipulation space, improper mapping between operator and robot motions, two-dimensional displays, cybersickness due to inconsistent

✉ Abdeldjallil Naceri
abdeldjallil.naceri@iit.it

[1] Istituto Italiano di Tecnologia (IIT), Via Morego, 30, 16163, Genova, Italy

[2] Munich School of Robotics and Machine Intelligence (MSRM), Technical University of Munich (TUM), 80797 Munich, Germany

[3] Mechanical Engineering and Materials Science, Yale University, Mason Lab, 9 Hillhouse Ave, New Haven, CT 06520, USA

[4] Dipartimento di Ingegneriadell'Informazione, Universit'a degli Studi di Siena, Via Roma 56, 53100, Siena, Italy

visual feedback, among others [5, 10, 20]. In contrast, modern virtual reality (VR) interfaces, owing to the video-gaming community, have seen important technological advances in graphics engines and devices [29] in terms of immersion, the development of natural physical control devices, high-fidelity graphical renderings, and native user viewpoint changes for near-natural visualizations (termed as "*teleporting*"), which overcome some of the limitations mentioned above.

During teleoperation, human operators are believed to build a mental model of the affordances of the task and the remote environment, based on the available visual feedback [14]. The features of VR can facilitate the building of such mental models as it can present the spatial information in 3D as compared to the traditional 2D displays. Presence (being there) and immersion are key factors of VR, characterized by the quality of the perceptual experience of the human operator [6, 36, 46, 47]. Greater immersion typically elicits a greater sense of presence [22, 51], and thereby improves performance in teleoperation [21]. Utilizing VR in robotics, therefore, seems logical, and has been recognized as such for its inherent benefits [3, 25, 40]. Milgram and Ballantyne [30] have highlighted the similarity between telerobotics and VR, sharing the common structure of *projecting* the user to the remote environment.

Building on these characteristics, this article presents a systematic approach for utilizing VR-based user interfaces in modern telerobotic applications in hazardous environments. The objective is to improve the immersion and presence in the remote teleoperation, and providing the *teleporting* feature to allow users to overcome visual occlusions in the VR scene. We hypothesize that these two factors significantly affect the performance and utility of the interface in the real-world. We present *Vicarios*, a VR-based interface that facilitates real-time remote robotic teleoperation , while exploiting the immersiveness and high-fidelity of VR for real-time feedback. *Vicarios* aims to keep the user's interaction natural and intuitive, by *projecting* the action to (and the perception from) the remote environment, as closely as possible. The central premise of *Vicarios* involves: (i) the stereoscopic virtual rendering of the real remote scene, including models of the remote robotic platforms, coupled with (ii) the real-time video streaming and depth information feedback from the remote environment; (iii) *teleporting* in the VR environment to have to have a better view of the teleoperation space and overcome occlusions; and (iv) mapping of the operator gestures/actions to the remote scene/devices relative to the actual user viewpoint. The integrated *Vicarios* interface, seen in Fig. 1, forms the basis of a framework that is extensible and adaptable to different teleoperation scenarios. The following sections outline the contribution of this research in detail. [1]

2 Related Work

The *Vicarios* interface draws on prior work in bridging VR with robotic systems and enhancing visual feedback from the remote environment. Researchers have long seen the advantages of using 3D graphics to overcome limitations of 2D displays, to better assist in robot motion planning and help with operator training [3, 30]. The more recent investigations in human-robot interaction have focused on using affordable VR devices (e.g., [25]), integrating the VR graphics engine softwares (e.g., Unity3D, Unreal) with compatible robotic hardware. Probably the most commonly used robot software is the Robot Operating System (ROS) [42]. Below, the most recent works in this field are discussed.

Peppoloni et al. [40], were one of the first groups to propose the concept of combining ROS-compatible robotic platforms and high performance VR interfaces for teleoperation. They combined the platforms with the Oculus Rift and the Leap Motion controller at the operator station, and the Kinect sensor for 3D point-cloud feedback from the remote scene. Subsequently, several research groups have explored VR-based teleoperation interfaces. A similar approach was adopted by [24] for a VR-teleoperation interface, using Unity3D, ROS, and ROSbridge for intercommunication. Several robot models were imported into the VR environment and interaction was implemented through gesture-tracking using the Leap Motion controller, mapping the motion in the virtual and real worlds. Lipton et al. [25] proposed a similar approach for bimanual manipulation, based on the Baxter robot, Unity3D, Razer Hydra hand trackers, and ROS. The user uses the VR world as the primary interface for interaction, and uses the real-time video stream from the real-world scene (shown in a customized window) for guidance. Another VR interface was introduced by [49] to allow users to interact with industrial robots via gestures, for trajectory planning. They allowed the users to define the robot trajectory in VR, in offline mode, to be executed by the real robot later, in a more supervisory form of interface. A more integrated approach was recently presented by Whitney et al. [50], where the imported robot meshes are rendered in Unity3D, whereas the remote scene and objects are rendered as a point-cloud from the remote depth camera. As visual feedback, the user views the robot mesh, the manipulated object from the point-cloud, and the customized video-feed received from robot-mounted cameras. Whitney et al. [50] present their innovation in importing different robot meshes and different mappings of the visualization and control of the robots.

As noted, combining ROS-enabled robotic platforms with VR graphics engines is quickly becoming a standard approach. Our contribution builds on the principles adopted

[1] A shorter version of this paper was presented at ICAR 2019 [34].

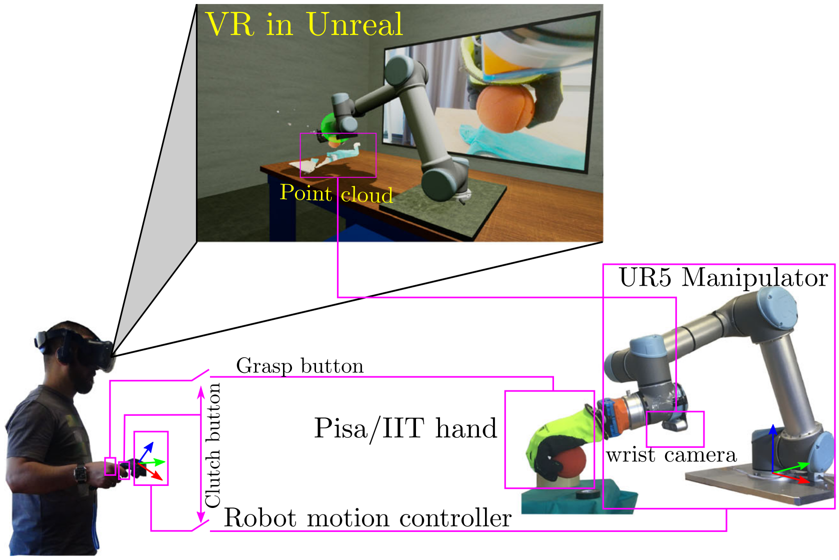


Fig. 1 The *Vicarios* Interface: A user teleoperating a remote robotic arm using the HTC Vive Pro VR system. Unreal VR graphics engine provides immersive visualization. Remote camera(s) provides video and depth feedback in real-time

by [24, 25, 40, 50], to advance the state-of-the-art in VR-based interfaces for teleoperation. A related aspect is that most of the above studies use cameras either mounted on the remote robot and/or fixed in the scene for a third-person view, which can be limited due to occlusions. To overcome this, Rakita et al. [44] propose the idea of multiple camera viewpoints in teleoperation using multiple cameras, drawing inspiration from the computer graphics domain, where moving the user viewpoint in a virtual scene is very useful in applications such as cinematic replays and people-tracking in crowd simulations [11, 17]. Our approach in this article is to integrate the native *teleporting* feature of VR to allow multiple viewpoints to the operator during tele-manipulation. Teleporting is a commonly used technique in VR for instantaneously changing one's position virtually [18], and VR users have shown improved performance with it [12, 26].

In this work, the *Vicarios* VR interface for remote robotic teleoperation is proposed, with the following contributions:

(i) One-to-one stereoscopic rendering of the remote environment, including models of the remote robotic platforms, in high quality VR for immersive visualization,
(ii) Real-time video streaming feedback, mono and stereo, for first-person views of the scene, and real-time point-cloud visualization for improved depth perception of the remote scene,
(iii) User viewpoint changes through teleporting in VR for intuitive visualization of the remote scene, and to overcome possible occlusions in the manipulation scene,
(iv) Viewpoint-independent mapping between operator gestures and remote robot motion.

The proposed interface is evaluated through a comparative user study, quantifying the effects of introducing such an interface on participants' performance in representative remote teleoperation functions. Specifically, for the immersion and presence factor, we test *Vicarios* against the traditional real-world stereo visual feedback condition (baseline). Further, we tested two conditions, with and without the teleporting feature, to establish the utility and impact of the feature on user performance. We used pre-defined user viewpoint positions to keep the goals simple at this initial validation stage. Finally, we discuss the extension of the *Vicarios* interface, integrating the teleporting feature with a moving remote camera, adapting it in VR as well as in the real-world. In this article, the terms describing the framework, experiments, and results, e.g.,, *tasks, goals*, etc., are in line with the IEEE standards referenced in [15, 37].

3 *Vicarios* Platform Description

The overall setup of the *Vicarios* interface, seen in Fig. 1, follows a general teleoperation setup, which includes an operator with a visualization interface and a remote environment. The proposed framework, as shown in Fig. 2, is divided into three major parts: the **operator site**, the **remote environment**, and a **communication network** between them. The gesture/motion controllers at the operator site convey the commands to the remote robots. The remote robots and the environment are rendered virtually in the VR-based visualization interface. The communication network allows real-time data exchange between the operator site and the remote environment, i.e.,, sending commands, receiving remote robot status, along with the real-time video and point-cloud information.

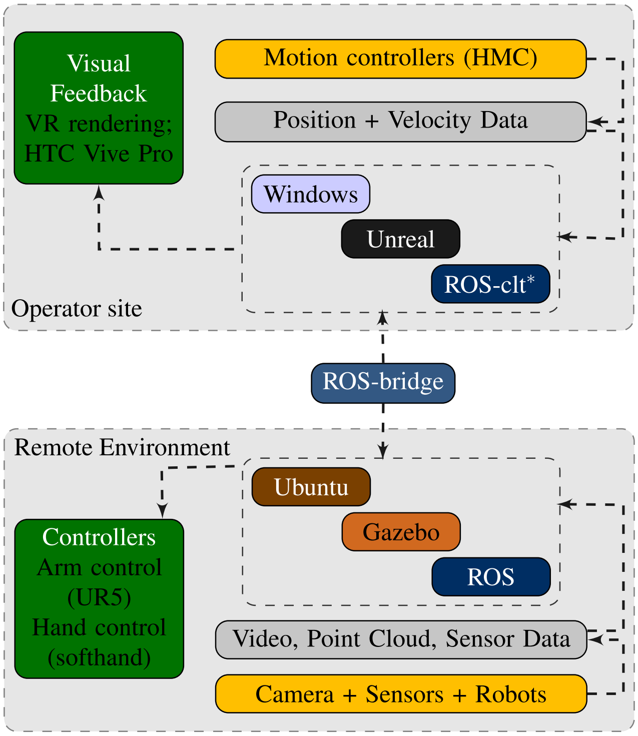

Visual Feedback VR rendering; HTC Vive Pro
Motion controllers (HMC)
Position + Velocity Data
Windows
Unreal
ROS-clt*
Operator site
ROS-bridge
Remote Environment
Ubuntu
Gazebo
ROS
Controllers Arm control (UR5) Hand control (softhand)
Video, Point Cloud, Sensor Data
Camera + Sensors + Robots

Fig. 2 Schema shows an overview of the *Vicarios* software and hardware framework. *ROS-clt: is ROSbridge client running under Windows at the operator site

3.1 *Vicarios* Platform Implementation

A practical implementation of *Vicarios* was realized, based on the schema detailed in Fig. 2. The overall components are seen in Fig. 1, and described in detail in the following subsections.

3.1.1 Operator Site

The operator site consisted of the following components:

a) Visualization hardware: an HTC Vive Pro head-mount display unit (hereinafter, referred to as HTC), with a display resolution of 2880 x 1600, a 110-degree field-of-view, and a 90 Hz refresh rate. It includes its own proprietary Lighthouse tracking system for accurate head-motion tracking.

b) Visualization software: the VR-based interface was based on the Unreal graphics engine on Windows 10. The VR scene in Unreal displayed the virtual 3D models of the remote robots, rendered 1:1 the remote environment virtually, and provided the real-time video feed (mono and stereo). Unreal permits importation, display and animation of the robot models, complete with their links and joints hierarchy. Robot models are imported as URDF (Universal Robot Description Framework) files, based on an open C++ library [2]. A dedicated robot message controller parses messages between the remote environment and the operator site, allowing real-time, 1:1 mapping between the motions of the real and the virtual robots.

c) Command interface: The operator commanded the remote scene using the HTC motion controllers (HMC), giving them a flexibility to teleoperate the remote robots in real-time using an ungrounded motion controller interface. The HMC are laser-based external tracking system for an accurate 6D pose estimation, within 2 mm of resolution, allowing the exploration of the visualized environment as well as natural interaction with the remote robots. There are three buttons on the HMC that the operator can use for:

   i) triggering the *clutch* action to engage / disengage the motion between the HMC and the remote robot. This overcomes the motion range limitation of the HMC as compared to that of the remote robot,

   ii) triggering the grasp action to open/close the remote robot hand end-effector,

   iii) triggering the user viewpoint changes in VR as desired by the operator for teleporting.

3.1.2 Remote Environment

The remote environment consisted of a Universal Robot UR5 manipulator, attached with a Pisa/IIT SoftHand [1]. The UR5 is a 6 degrees-of-freedom (DOFs) manipulator, able to move at up to 1 m/s and each joint at 180 deg/s. It has a payload of 5 kg and a repeatability of $\pm 0.1mm$. The Pisa/IIT SoftHand is an underactuated soft robotic hand with 19 joints, but uses only one actuator to activate its adaptive synergy [4, 9]. The actuator is driven by commanding a grasping variable $g_r \in [0, 1]$, which drives the hand to open or close.

For visual feedback, up to three cameras were mounted in the remote environment, streaming real-time video and point-cloud data, including: (i) one ZED-mini stereoscopic RGB-D camera (2560x720 @ 60 Hz; depth range 0.1 - 15 m) for 3D video feed and point-cloud data; and (ii) two Intel Realsense RGB-D (1280 x 720 @ 90Hz; depth range of 0.2 - 10 m) cameras, one mounted on the UR5 wrist for the close-up video feed and one external for point-cloud data.

The remote environment also consisted of a mounting bench for the robot platform. As a use-case in this article, a sphere (tennis ball) and a cylindrical base are also introduced for pick-and-place tasks.

3.1.3 Communication Network

A direct Ethernet LAN connection was used between the remote environment and the operator site. Robot Operating

System (ROS) [42] served as the framework between the operator VR interface and the remote robots, implemented in Ubuntu 16.04. Unreal on Windows communicated with the remote environment devices on Linux via UDP sockets and ROSbridge [13].

Within Unreal, starting from the open source C++ library [2], the inherent ROS architecture was encapsulated in dedicated classes and Unreal Blueprints, making it more convenient to connect to a ROS server, to add ROS topics' publishers or subscribers, and to add/modify ROS messages and types. The UDP sockets allowed the streaming of video-feed and point-cloud data from the remote environment. A distinct advantage of using VR as an interface is that it does not require more communication bandwidth than its traditional counterparts (e.g., 2D monitors, mouse / keyboard input, etc). The processing of the scene-rendering and meshes is done at the operator site itself, without relying on any communication channel.

Communication (re-)establishment The communication between the client (operator site) and the server (remote environment) is established using a handshake protocol via ROSbridge. The protocol involves probing the accessibility to the server ROS topics at the client, thereby confirming if the client-server are connected or not. As soon as the probe is successful, i.e.,, connection is established, the user is notified (through a visual cue) and their commands can be sent to the remote robot via ROSbridge. In case of a loss of communication or an unsuccessful handshake, the user is notified via a warning cue and all command topics are set to default values (e.g., zeros) to avoid any hiccups in robot motion. The loss of communications ccould be a hardware (cables disconnected) or software problem. The handshake protocol is re-triggered as above to re-establish communication. When the problem persists, the complete hardware setup is then investigated as part of the solution to the problem.

Data Streaming Real-time, high resolution video and point-cloud data requires a high bandwidth and can suffer from latency issues. To address this, we developed a real-time video and point-cloud streaming system, that has five key components: (i) data acquisition using the software drivers of the respective cameras, (ii) an encoder that compresses the acquired data (point-cloud and video) from the cameras in the remote site, (ii) a streaming algorithm to send the data over the network, (iii) a decoder that decompresses the data received at the operator site, and (iv) a rendering system that plays the incoming streams.

The video streaming system was developed based on the FFmpeg multimedia framework, with the H.264 video compression and the Real-time Transport Protocol (RTP) for streaming. The point-cloud streaming system included a state-of-the-art encoding algorithm [27], and used the Boost ASIO over a TCP socket for streaming.

4 Visualization and Viewpoint-independent Motion Mapping

The *Vicarios* interface aims to provide smooth visualization and intuitive motion mapping for seamless coordination between the operator and remote sites.

4.1 Visualization

The visualization in the HTC of the remote scene is shown in Fig. 1. The flexibility of the VR environment is evident in the arrangement of the remote scene and the placement of the camera-feed and the point-cloud data. The VR environment can closely replicate the remote scene based on the information available about the remote environment. Depending on the application, this knowledge can be obtained either through *a priori* scene layout information and/or real-time capture, reconstruction, and rendering in VR of the 3D remote environment. The scene needs to be rendered in a way that the operator feels immersed in it and can focus on performing the task.

Vicarios uses a combination of *a priori* scene knowledge and real-time scene capture of the remote environment for the creation of the VR environment. The *a priori* knowledge, including the known dimensions, kinematics, and locations of the remote robot platform, the poses of the RGB-D cameras, and the mounting bench (seen in Fig. 1), helps to locate the corresponding mesh models in the VR scene. The motion of the robot model in VR is animated using the real-time joint-states information from the real robot via ROSbridge (noted in Section 3.1.3). The dynamic elements, e.g., those involved in the pick-and-place task, the sphere and the cylindrical base, are not known *a priori*, and are obtained through real-time capture from the remote environment using the camera feeds and point-cloud data. The point-clouds are projected in an intuitive manner within the VR scene, registered to the *a priori* known pose of the RGB-D cameras. The camera intrinsic parameters are used for the registration, and hand-eye calibration is performed to overcome any misalignment due to inaccuracies in parameters. The video-feeds are displayed on embedded screens positioned within the VR scene. This mixture of virtual models and the real-world point-clouds is termed as *augmented virtuality* with the objective of allowing users to infer the relative pose of those objects in the remote environment with respect to the actual robot pose.

Therefore, in its present use-case scenario, *Vicarios* renders the models of *a priori* known objects from the remote environment that are presumably structured and fixed in terms of location. The novel contribution here is the projection of the live point-cloud data within VR, which provides the 3D pose and dimensions of the manipulated objects. This facilitates the users in combining the information from the real and virtual worlds within the same visualization paradigm, thereby enhancing the sense of immersion and presence.

4.2 Motion Mapping

As mentioned earlier, the operator commands the remote scene, i.e.,, the robot motion, directly using the HMC. The conventional *sense-plan-act* control paradigm [8] therefore, is based on a human-in-the-loop design, without any assumption of autonomy (or semi-autonomy) on part of the robots. The sensed data (i.e.,, video, point-cloud, etc.) from the scene is communicated to the human operator in real-time, based on which the operator plans the next step, and in-turn sends the commands for action to the robot. At this stage, the remote robots are assumed to be *passive*, i.e.,, not acting autonomously. Based on the control algorithm of choice for the robots (e.g., Jacobian, Inverse Kinematics), the calculated joint angles, velocities, and accelerations are sent to the remote robots from the HMC interface. The joint-states of the remote robots are, in turn, communicated back to the interface to update the poses of the rendered robot models. This integration means that the operator actually visualizes and commands the motion of the virtual robot. The real robot's motion is mapped 1:1 to this virtual robot motion.

Since the kinematics of the remote robot and the operator controllers are non-homothetic, the 6-DOF pose of the robot is commanded using velocity-control, i.e., the velocity of the HMC is mapped to the robot, calculated using the inverse Jacobian method. The robot pose is commanded by finding the position error $\boldsymbol{e} \in SE(3)$ between the current and the desired robot pose. Then, the damped least-squares solution to equation (1) iteratively finds the change in joint angles $\Delta \boldsymbol{q}$ that minimizes the error $\boldsymbol{e}$ [7].

$$\Delta \boldsymbol{q} = (\boldsymbol{J}^T \boldsymbol{J} + \lambda \boldsymbol{I})^{-1} \boldsymbol{J}^T \boldsymbol{e}, \tag{1}$$

$\boldsymbol{J}$ is the 6-DOF jacobian of the robot, and $\lambda \in R$ is a non-zero damping constant. Finally, a proportional controller $\dot{\boldsymbol{q}} = K_p \cdot \Delta \boldsymbol{q}$ sets the joint velocities to reach the desired pose. The values for λ as 0.001 and K_p as 0.6, were empirically obtained.

4.3 Viewpoint-independent Motion Mapping

The aforementioned approach also grants viewpoint-independent mapping between the gestures of the HMC and the motion of the remote robot. This is an important feature facilitated by the *Vicarios* interface, to let the operator freely change the viewpoint within the VR interface, without worrying about re-mapping their gestures. It is hypothesized (proven later in the text) that the ability to observe the remote scene (virtually) from different perspectives improves the accuracy when doing complex remote telemanipulation.

To allow this mapping, the following method is adopted. When the *clutch* button of the HMC is pressed, i.e., the motion between the HMC and the remote robot is engaged, the pose of the HMC, $\boldsymbol{h}(t_0)$, and the robot, $\boldsymbol{r}(t_0)$, are saved. When the user moves their hand to new a pose $\boldsymbol{h}(t_1)$, the difference between the $\boldsymbol{h}(t_0)$ and $\boldsymbol{h}(t_1)$, i.e., $\Delta \boldsymbol{h}$, is calculated. $\Delta \boldsymbol{h}$ is then transformed to the robot base frame, scaled, and added to $\boldsymbol{r}(t_0)$ to obtain the desired new robot pose (corresponding to $\boldsymbol{h}(t_1)$). The equations are as follows:

$$\begin{aligned} \Delta \boldsymbol{h}^o &= \boldsymbol{h}^o(t_1) \boldsymbol{h}^o(t_0)^{-1} \\ \Delta \boldsymbol{h}^p &= \boldsymbol{h}^p(t_1) - \boldsymbol{h}^p(t_0), \end{aligned} \tag{2}$$

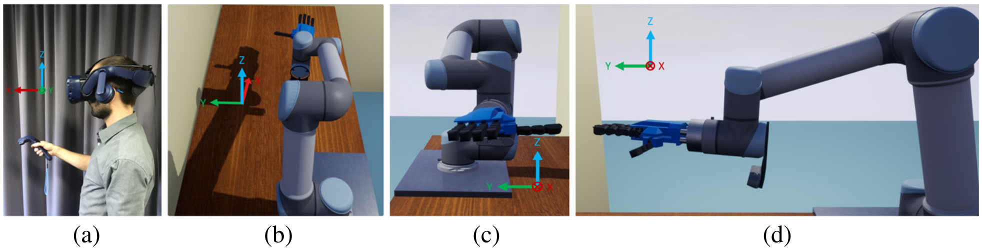
where $\{\cdot\}^o \in \mathbb{R}^{3\times 3}$ is the orientation matrix and $\{\cdot\}^p \in \mathbb{R}^{3\times 1}$ is the position vector. The mapping to the robot frame is done by adding the desired transformation ${}^W\Delta \boldsymbol{h}$ in the global frame W to the robot's current pose $\boldsymbol{r}(t_0)$, obtaining the new pose $\boldsymbol{r}(t_1)$:

$$\begin{aligned} \boldsymbol{r}^o(t_1) &= {}^W\boldsymbol{R}_R^{-1} \cdot {}^W\Delta \boldsymbol{h}^o \cdot {}^W\boldsymbol{R}_R \cdot \boldsymbol{r}^o(t_0) \\ \boldsymbol{r}^p(t_1) &= \boldsymbol{r}^p(t_0) \cdot \left({}^W\boldsymbol{R}_R \cdot \boldsymbol{r}^p(t_0)\right) \cdot {}^W\Delta \boldsymbol{h}^p \text{ ,} \end{aligned} \tag{3}$$

where ${}^W\boldsymbol{R}_R$ is the rotation matrix from the world frame W to the robot frame R. Figure 3 demonstrates the concept with sample user motions mapped to the robot according to the chosen user viewpoint, without requiring the user to re-orient their gestures/hand movements. The viewpoints can have different positions and orientations in 3D space.

The above-mentioned world and robot frames are both in the Unreal engine. Commanding the real robot implies the robot coordinate frame in Unreal needs to be transformed to that of ROS. Due to the differences in the way the Unreal and ROS platforms have been developed, for positions, the right-handed coordinate frame of ROS needs to be transformed to a left-handed one for Unreal, with a scaling factor of 100 for dimensional conformity. Similarly, for orientations, the $x-$ and $z-$ axes of ROS need to be inverted for Unreal.

Although velocity-based teleoperation allows real-time control between non-homothetic devices, it limits the range of motion to that of the motion control device, e.g., the HMC. This issue is easily solved by using the *clutch*-based system, mentioned earlier. The teleoperation control between the master device and the remote robots can be coupled and decoupled as required to change the workspace of the user. This allows the user to pause and resume the robot control as desired, avoiding complicated gestures/hand movements and providing greater user comfort.



(a) (b) (c) (d)

Fig. 3 Viewpoint-independent motion mapping - user motions are mapped to the robot motion independent of the chosen user viewpoint. The mapping is transparent to the operator. As an example, the $x-$axis in **a** is aligned with the user's viewing direction, going away from the viewing plane. This property is maintained even as the views change, i.e., the $x-$axis is aligned with and directed away from the viewing plane. **a** User reference **b** Motion reference in top view **c** Motion reference in front view **d** Motion reference in side view

5 *Vicarios* Evaluation - User Studies

The aim with *Vicarios* is to establish it as an immersive interface for remote teleoperation in unstructured environments. Towards this end, user studies were conducted to understand the performance and utility of the interface, and to test the effectiveness of the different features, including teleporting and viewpoint-independent motion mapping. In particular, we performed hypothesis testing to assess the impact on the user, positive or negative, of the *Vicarios* VR-based interface, against the traditional video-only interface. Below, we present the details of our experimental procedures, conditions, hypotheses, and analysis metrics.

5.1 Experiment Procedure

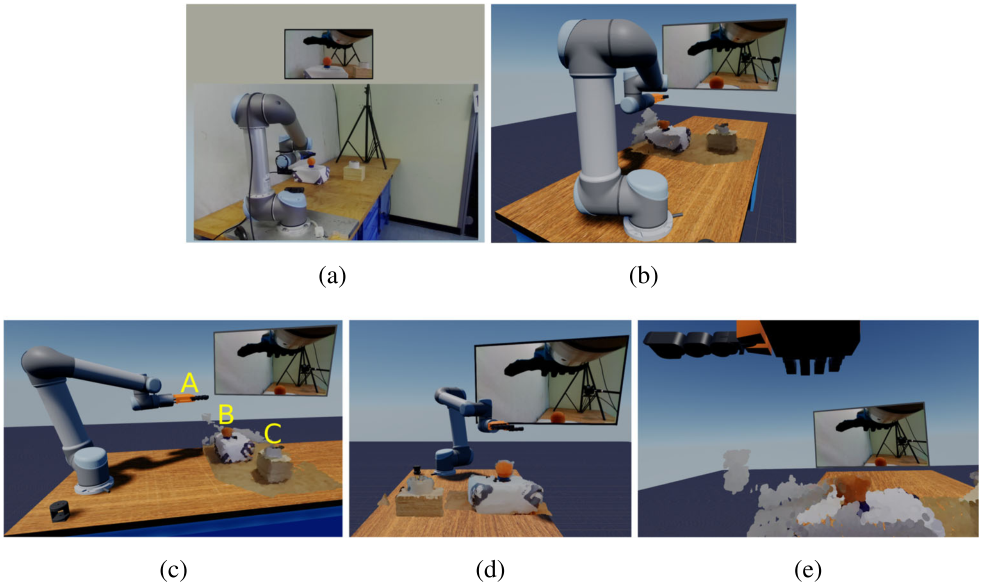
For the evaluation studies, as stated earlier, a tennis ball and a cylindrical base were used for pick-and-place tasks. The real-time video feed was displayed on virtual screens positioned within the VR environment. The point-cloud data was projected in the VR scene in the appropriate pose, referenced by the real-world camera poses.

The participants sat on a chair in a different room than the remote environment site and controlled the robot end-effector motions using the HMC. Participants viewed the remote scene through the HTC that was fixed (as seen in the supplementary video). This was done to have consistency across the different testing conditions, having no head movements in any of them. As stated earlier, we used a pick-and-place task to evaluate the interface. A fixed stereo camera, located behind the robot base served as the default user viewpoint, and allowed assessing our proposed teleporting feature in the presence of occlusions or obfuscations. At the "Go" signal from the experimenter, starting from point "A", the participant picked up the sphere (tennis ball) from point "B" and placed it inside a cylindrical base located at point "C" (Fig. 4c). Participants released the grasped sphere, based on their judgement of the end-effector location vis-a-vis the target location. At the end of each trial, the experimenter gave a "relax" signal to the participants, indicating to bring their head back from the HTC. At this stage, the robot arm was reset to its initial pose. The sphere and cylindrical base were replaced for the next trial; their locations were randomized across trials (Fig. 5c).

5.2 Experimental Conditions

For a comparative evaluation, the trials were performed in three different conditions:

1. *Real-world stereo (RW-Stereo)*: (baseline condition), participants performed the experiment viewing the remote setup through the real-time stereo video from the ZED-mini camera (seen in Figs. 4a, 5a) and the wrist-mounted realsense camera, without any VR rendering. Participants were limited to the viewpoints based on those two cameras only, simulating the more traditional teleoperation interface. Participants used the HMC to command the robot, using the clutch and grasp trigger buttons only.
2. *VR without teleporting (VR-No Teleport)*: participants were asked to perform the experiment, visualizing the remote scene through the *Vicarios* interface, but with a fixed user viewpoint (seen in Fig. 5b). The viewpoint for this condition was set the same as that used in the *RW-stereo* condition. Only the point-cloud data from the ZED-mini camera was shown in the VR scene (Figs. 4b, 5b) to render the objects in the scene. Participants also viewed the video feed from the wrist-mounted realsense, as in the baseline condition. Here too, only the clutch and grasp trigger buttons on the HMC were used.
3. *VR with teleporting (VR-With Teleport)*: In this condition as well, participants performed the experiment, visualizing the scene through the *Vicarios* interface. Participants used all 3 buttons on the HMC, and had the ability to change their current viewpoint, using teleporting, among four different view poses (Figs. 4b, c, d, e, 5c). At the beginning of each trial, they were asked to teleport through all 4 poses in order to remind



(a) (b)

(c) (d) (e)

Fig. 4 Experimental setup, VR scenes, and participants' viewpoints in different conditions. The interface includes virtual models of real-world robots, real-time video, and depth (point-cloud) data from the real-world. **a** Real-world stereo camera view rendered through the HTC. **b** Default VR view in *Vicarios* with stereo camera view on a panel, in the no teleporting condition. **b**, **c**, **d**, **e** Different viewpoints in the VR scene, as seen in the teleporting condition. In panel **c** "A" refers to starting location, "B" refers to picking location and "C" refers to placing location

them of the different viewpoints possible with this feature. Moreover, they were free to choose any viewpoint location at any time within the trial to perform the experiment. To cover for all the 4 viewpoints, in addition to the point-cloud data from the ZED-mini, which served user viewpoints $tp_{1,2,4}$ (Figs. 4b, c, e, 5c), a second realsense camera was added in the remote scene for tp_3 (Figs. 4e, 5c). This ensured the availability of the point-cloud data at every viewpoint during teleporting.

Each participant repeated the experiment ten times in only one assigned condition. At the end of the experiment, participants filled in a post-experiment questionnaire, rating certain pre-determined properties of the proposed interface on a 7-point Likert scale (1: bad and 7: good).

5.3 Participants

Twenty-four participants (Age: 30 ± 7, five females) took part in our evaluation study, divided into 3 groups of 8 each for the 3 conditions (a between participants design). All participants were right-handed and were naive to the study, having no prior experience with the *Vicarios* platform.

5.4 Hypotheses

We have two hypotheses: $\boldsymbol{H}_0$: we predict that participant performance in the *RW-Stereo* condition would be better than the *VR-No Teleport* condition due to the differences in depth cues (e.g., perspectives, lighting, shading etc) between the real-world and VR. It is usually difficult to reproduce exactly the same effect with the cues in VR as in the real-world. Any mismatch in depth cues between the real-world and VR alters participants' depth perception and reduces their performance in VR [35].

$\boldsymbol{H}_1$: we predict that participant performance would significantly improve in the *VR-With Teleport* condition, and would be better than the *VR-No Teleport* condition, owing to the teleporting feature that allows participants to overcome occlusions and/or obfuscations.

5.5 Analysis Metrics

We quantified participants' experiment completion times and their success rate as dependent variables of the experiment in all conditions. The success rate is defined as a percentage, counting the number of trials where the participants successfully picked the sphere and placed it in the target location, inside the cylindrical base. Further, we also analysed participants' ratings from the post-experiment questionnaire for certain properties of *Vicarios* (defined in Fig. 7).

6 Results

6.1 Performance and Completion Times

For the success rates, results of the user studies, as seen in Fig. 6a, revealed that participants recorded better values in the *VR-With Teleport* condition as compared to the other two conditions, i.e., *RW-Stereo* and *VR-No Teleport*. Specifically, the average success rate, calculated out of a total of 80 trials per condition, was 62.5 ± 14.90 (M ± SD %) in *RW-Stereo* . It dropped to 51.25 ± 19.60% for *VR-No Teleport*, but reached 90.00 ± 5.30% for the *VR-*

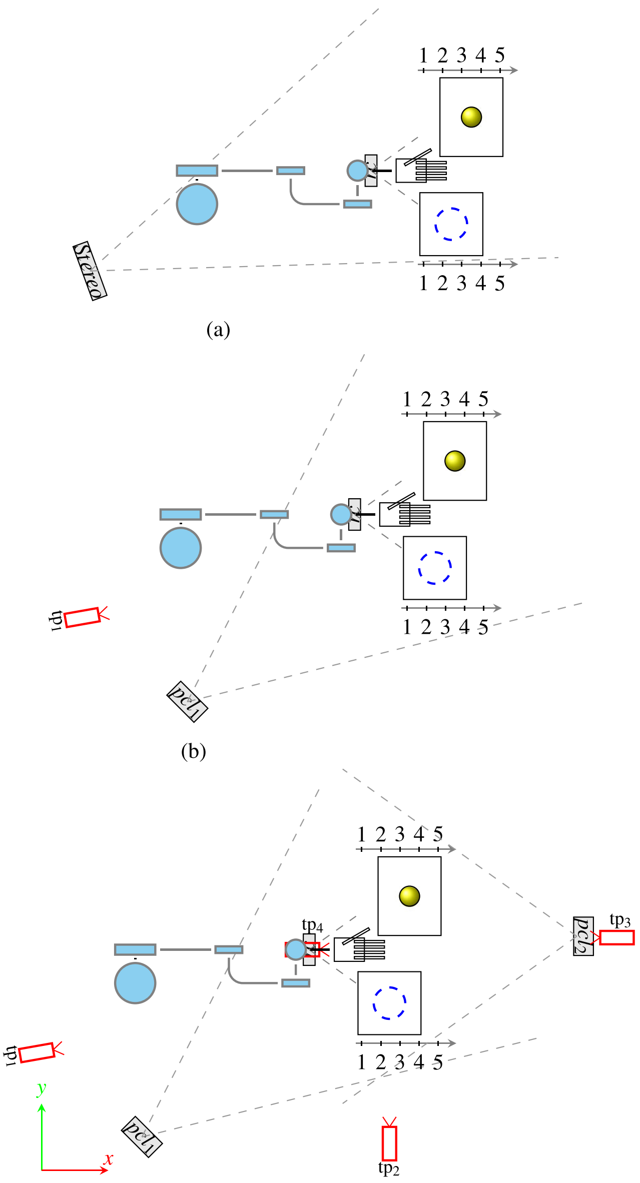


Fig. 5 Schematic of the experimental scene in different conditions. Yellow circle represents the manipulated sphere (tennis ball) in its initial location. Blue dashed circle represents the target final location (cylindrical base). Positions 1-to-5 show the possible locations of the initial and target locations that were randomized across trials for each participant. **a** *RW-Stereo* condition setup with the ZED-mini camera (Stereo) and UR5-wrist-mounted realsense camera (rc) for video-feeds. **b** *VR-No Teleport* condition with pcl_1 point-cloud data coming from ZED-mini camera, rc for video-feed and tp_1 represents user viewpoint location in VR environment. **c** (*VR-With Teleport*) condition with $\text{pcl}_{1,2}$ locations of the ZED-mini and realsense cameras in remote scene used to project point-cloud data. $\text{tp}_{1,2,3,4}$ represent the teleport locations

With Teleport. Moreover, the one-way analysis of variance (ANOVA) on participants' scores revealed a significant difference in the means of the group conditions ($F(2, 21) = 15.04$; $p < 0.001$). A Tukey's test revealed a significant difference of pairs: *VR-With Teleport* vs. *VR-No Teleport* ($p < 0.001$) and *VR-With Teleport* vs. *RW-Stereo* ($p < 0.01$) but no significant effect between *RW-Stereo* and *VR-No Teleport* ($p = 0.29$). In other words, this implies that our hypothesis $\boldsymbol{H}_0$ is rejected, while $\boldsymbol{H}_1$ is approved. The results indicate that multiple user viewpoints included in the last condition contributed to participants achieving the pick-and-place task more accurately. Indeed, participants teleported at least twice in the third condition in order to get a better viewpoint, especially when the manipulation scene was occluded. A closer analysis revealed that participants used viewpoints 2 and 3 ($\text{tp}_{2,3}$ Fig. 5a) the most.

For the completion times, seen in Fig. 6b, participants recorded 18.98 ± 8.60 sec. (Mean ± confidence intervals) during *RW-Stereo* condition, 16.75 ± 7.00 sec. in the *VR-No Teleport* condition, and 29.15 ± 14.31 sec. in the *VR-With Teleport* condition. Evidently, participants were slower in the third condition, requiring more time when using the teleporting feature.

6.2 Post-Experiment Questionnaire Results

Results of the post-experiment questionnaire, seen in Fig. 7, revealed the following:

(i) the *VR-With Teleport* condition received the highest ratings for the properties of 'helpfulness of visual information' and 'sufficiency of viewpoint(s)', while the *RW-Stereo* condition receive the lowest. This result is in accordance with the reported results for the participant performance.

(ii) Participants reported that the task was more physically demanding in the *VR-With Teleport* condition , while the *RW-Stereo* condition was rated the least demanding.

(iii) For 'setup affordance' (ease-of-use) and 'satisfaction with the pace of task completion', participants rated the *RW-Stereo* condition better than the VR conditions.

(iv) For the 'control intuitiveness (ease-of-control of remote robot)' item, the *VR-No Teleport* condition was rated lower than the other two conditions, while for 'task easiness', it was rated better than the other two.

(v) As seen in Fig. 7, the rest of the questionnaire items were rated quite similarly across conditions, including 'confidence (with the setup/task)', 'robot pace', 'cyber-sickness', 'stress', and 'hurriedness'.

7 Discussion

In this paper, we investigated the features and performance of the *Vicarios* interface against two hypotheses, $\boldsymbol{H}_0$ and

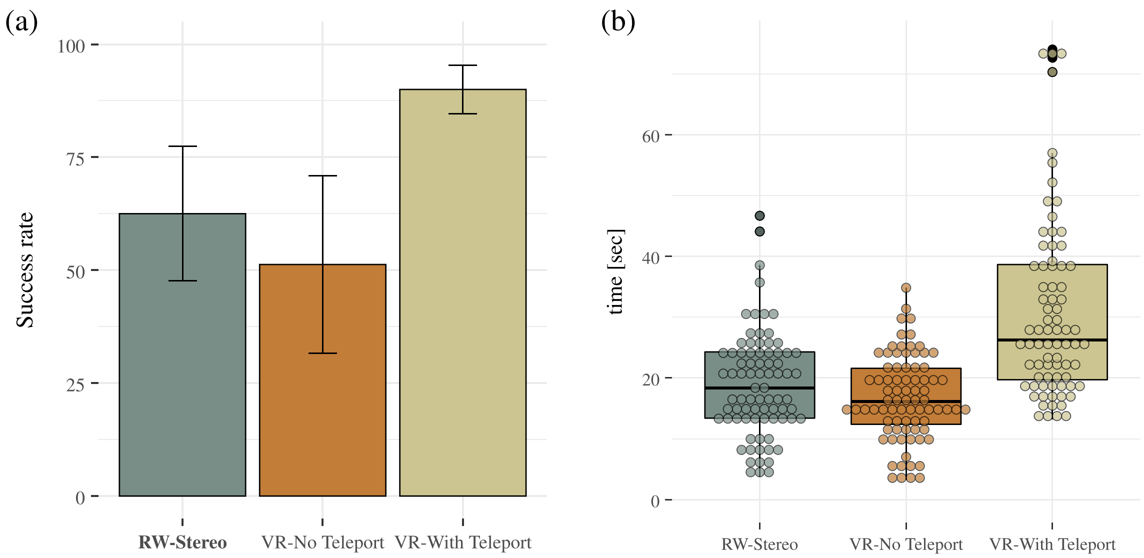


Fig. 6 Participants' performance results in different conditions. **a** Average success rates in all conditions. Error bars represent the standard deviation. **b** Boxplot for completion time in different conditions. Dots represents individual trials

$\boldsymbol{H}_1$. The results were conclusive to outright reject $\boldsymbol{H}_0$ and approve $\boldsymbol{H}_1$.

7.1 Performance Analysis

7.1.1 H_0 Rejected

Testing the hypothesis for the participants' performance in the *RW-Stereo* and *VR-No Teleport* conditions, it was observed that they were *not* significantly different. The task completion times recorded in the conditions were also not significantly different. This led to the rejection of $\boldsymbol{H}_0$. This result was unexpected, especially since previous studies have showed that users' performance using stereo camera views from the real-world is much better than through a VR interface, due to the diminished quality of the depth cues and realism in VR [35]. The rejection of $\boldsymbol{H}_0$ might indicate that the *Vicarios* interface VR environment is able to match the real-world scene, viewed through the stereo camera. The authors of [48] proposed a similar immersive VR-based robot teleoperation system, and compared it to a purely video-based teleoperation system viewed on large screen. User performances and the post-experiment questionnaire data credited the VR-based approach more than the video-feed condition. This result is along expected lines, attributed to the lack of immersion in the video-only condition. In our case though, it is probable that the scene layout reproduced in the VR condition of *VR-No Teleport* is identical to that seen in the *RW-Stereo*. This result indicates that the immersion variable in *Vicarios* was adequately satisfied in the VR condition. This relates to the concepts of immersion and presence, where immersion can be defined as the technological sophistication of a particular VR system, while presence can be defined as its perceptual counterpart. More immersive technologies typically elicit a greater sense of presence from the users [22, 51]. Participants' presence factor can be increased dramatically by increasing realism of the VR depth cues, e.g., realistic lighting, perspectives, textures, shadowing, etc. [33, 45]. In fact, it has been shown that participants' performance depends on the quality of immersion and presence in VR [6]. Therefore, it can be concluded that the depth cues reproduced in VR were realistic and close to those viewed in the real-world condition (*RW-Stereo*).

7.1.2 H_1 Approved

The hypothesis tested whether the ability to change one's viewpoint in the *VR-With Teleport* condition, using the teleporting feature, would improve the participants' performance compared to the *RW-Stereo* and *VR-No Teleport* conditions. The feature allowed teleporting to predefined positions within VR and provided relative motion mapping for intuitive user gestures. The reason for using predefined teleporting poses was to make it simple and easy-to-use. The overall results demonstrated that our hypothesis $\boldsymbol{H}_1$, was approved, confirming that this added feature significantly improved participants' performance. The lower success rates recorded for the other conditions were mainly due to the occlusions caused by the moving robot in the manipulation scene. Indeed, in the *VR-With Teleport* condition, participants used poses 2 and 3 ($\text{tp}_{2,3}$ in Fig. 5c) intensively to overcome the occlusions when moving the robot arm, to better visualize the target scene.

However, the task completion times recorded in the *VR-With Teleport* were longer by nearly 10 sec. as compared to

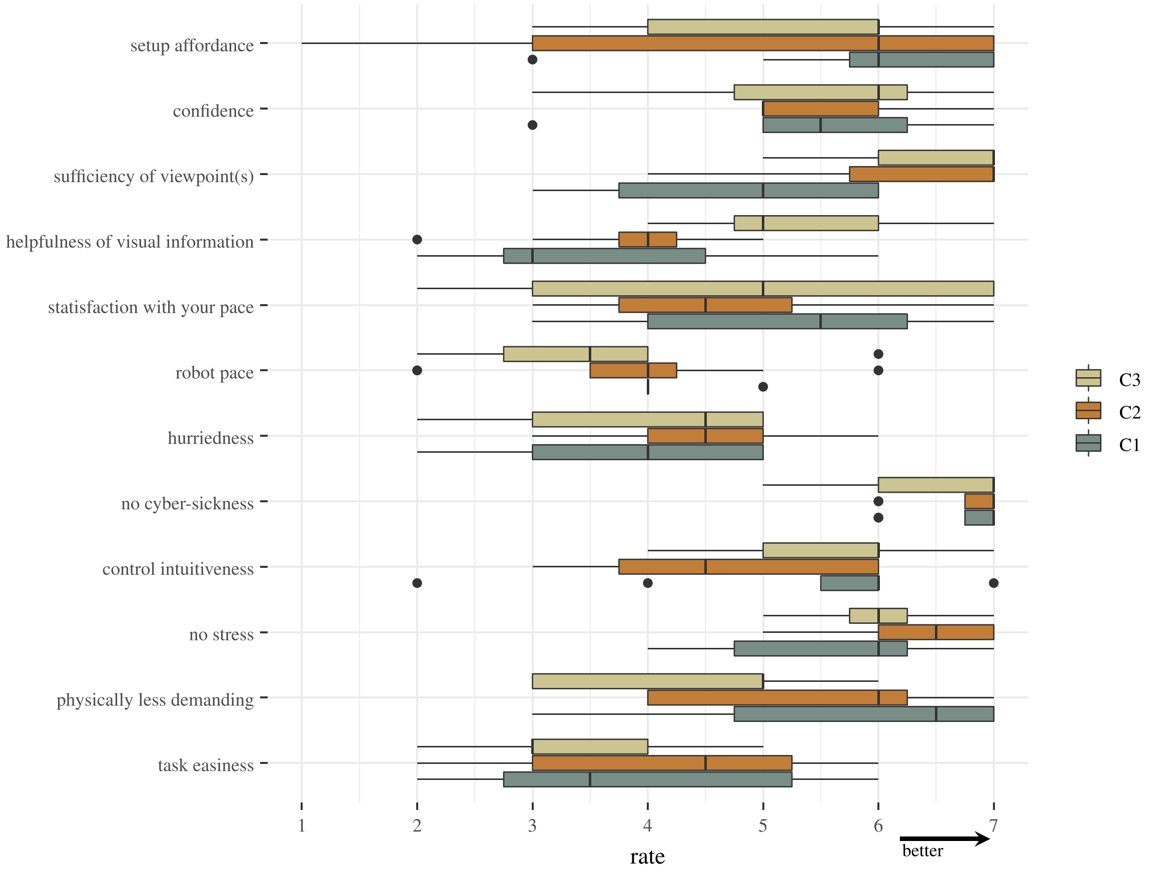


Fig. 7 Barplot for post-experiment questionnaire data. Black dots represents outliers. **C1**: RW-Stereo, **C2**: VR-No Teleport, **C3**: VR-With Teleport

the other two conditions. This was mainly due to the fact that participants teleported more than once across the four poses in the VR scene ($tp_{1,2,3,4}$: Fig. 5c).

7.2 Latency Analysis

7.2.1 Motion Mapping Latency

To evaluate the latency in the motion mapping Eqs. 2 and 3, the setup was controlled for a pre-defined period of time. A cross-correlation analysis was done between the user's Cartesian command motions (recorded from the HMC) and the UR5 end-effector Cartesian motions. The robot was set to full speed (i.e., K_p to 1.0). The measured round-trip latency between the user commanding the motion and the user visualizing that motion on the robot in VR is 80 ms, implying a 40 ms one-way latency between user command and actual robot motion. This takes into account the VR command transmission time, the robot arm controller loop, the overall network communication between the remote site and the operator site (with ROSbridge), and the rendering time for the robot in VR.

7.2.2 Data Streaming Latency

As noted earlier, bandwidth and latency are key issues when using streaming point-cloud and video data over network. We implemented a latency measurement tool to capture the latency of the data streams. This included three parts: (i) a Chrono high-resolution clock to measure the execution time of each of the five components involved in streaming, mentioned in Section 3.1.3, (ii) the network time protocol (NTP) to synchronize the computers at the remote site and the operator site, and (iii) a barcode mechanism with frame-id and time for each acquired video frame, and a timestamped header for the point-cloud frame. At the operator site, the barcode is used to synchronize the received/decoded stream and to measure latency over the network. Keeping an HD resolution (1280 x 720) at 30 fps, the following one-way, end-to-end (from remote -to-operator) latency was measured:

- ZED-mini stereo video: 180ms,
- Realsense video: 110 ms,
- Point-cloud (Either camera, 2m depth, 921600 encoded points): 600 ms

The values for all three data streams are within those observed in literature for state-of-the-art systems [19]. The questionnaire analysis also shows that these latency values do not have an adverse impact on participants in terms of cyber-sickness or control intuitiveness. For *Vicarios*, end-to-end latency involves processing times in different stages, e.g.,, image / point-cloud acquisition, data compression, streaming, data decoding, and display rendering. A deeper analysis of the above end-to-end latency values is important to isolate and improve the components in order to optimise the overall latency. This problem is being currently addressed and under investigation.

7.3 Questionnaire Analysis

For the post-experiment questionnaire, here we express the questions where significant differences were recorded among the conditions.

- Participants rated the *VR-With Teleport* condition highly for the questions concerning viewpoints and felt that the available user viewpoints were more helpful, as compared to the other two conditions. For *sufficiency of viewpoint(s)*, they rated the *RW-Stereo* condition the lowest.
- Participants found the tasks to be more physically demanding (*physically less demanding*) and more difficult (*task easiness*) in the *VR-With Teleport* condition, as compared to the other two conditions, although the experiment across conditions was the same, except the teleport feature added in the third condition. Recall that the only difference here is the addition of a button for changing the user viewpoint. This is explained in light of previous studies, which show that teleporting techniques can create cognitive overload on participants [38]. On the other hand, it is clear that teleporting motivated participants to be more careful during task execution, which resulted in higher success rates for the *VR-With Teleport* condition.
- In all 3 conditions, participants expressed their satisfaction with the affordances of the setup (ease-of-use) and felt confident in using the interface (*setup affordance* and *confidence*).
- The robot control was rated to be similarly intuitive (*control intuitiveness*) for the *RW-Stereo* and *VR-With Teleport* conditions, while the *VR-No Teleport* condition was rated lower. We recall that this is a ‘between participants’ study. This result, therefore implies that the introduction of *Vicarios*, especially with teleporting, permits the participants to control the remote robot just as intuitively and transparently as they do without VR. This detail follows from the earlier discussed rejection of hypothesis $\boldsymbol{H}_0$ as well.
- In all 3 conditions, participants were satisfied with their pace (*satisfaction with your pace*) and felt that the robot was slow (*robot pace*). Indeed, the robot was run only at 60% speed during the experiment for safety reasons. This is not a hard constraint and can be removed as desired depending on user’s expertise.
- For *hurriedness*, participants gave average rating in all 3 conditions, implying that the visualization interface was not a factor in their sense of hurriedness in task completion. On the other hand, they felt comfortable in using the VR-based interface (*no cyber-sickness* and *no stress*).

Overall, based on the participants’ performance and the post-questionnaire results, the *Vicarios* interface is evaluated positively, including for the teleporting feature. Nevertheless, one pertinent question is on the placement of the remote cameras. Here, we used only 3 fixed camera locations, but the locations might differ depending on the goal and/or the remote environment itself. Put another way, how realistic would the teleporting feature be in a real teleoperation scenario and whether it would be possible to extend this approach so that the remote camera locations can be easily changed to adapt to the user’s desired viewpoint, the goal requirements, and the cluttered remote environment? These questions are addressed in what is discussed next.

7.4 Teleporting in VR-based Teleoperation with a Motion-Capable Remote Camera

In the current study, the *VR-With Teleport* condition used 2 fixed external remote cameras, the ZED-mini and the realsense, and 1 wrist-mounted camera (Fig. 5c) to receive a sufficient perspective of the remote scene in the video and point-cloud data. Participants were asked to teleport only to predefined user viewpoints (four possible locations) in order to keep the function simple and to better control the experimental variables. The results from the task performance and the questionnaire provided some vital clues as to how the practical setup for this condition in *Vicarios* could be improved. Outside the VR context, the authors of [44] developed a method to improve the traditional teleoperation visual feedback by providing users with an effective real-time camera viewpoint using a second camera-in-hand robot arm. In the case of *Vicarios*, the viewpoint-independent motion mapping feature potentially allows a user to choose any user viewpoint in VR to visualize the remote scene. The limitation here would be the different camera viewpoints available for visual information coming from the real-world, i.e., videos and point-clouds. Taking inspiration from [44] towards integrating the changing camera viewpoint idea, we propose the following robotic system that allows for the teleporting feature in a plausible teleoperation scenario Fig. 8. The part that is added to the *Vicarios* platform includes a robotic arm (Franka Emika), with the external ZED-mini mounted on its end-effector. This makes the ZED-mini motion-capable (the *camera-robot*), allowing for changing its viewpoints in the remote scene, and removing the need for the external realsense. The teleport locations can be easily modified based on the requirements. Moreover, different user modes (e.g., expert vs. regular) can be implemented, where only certain users can control the moving camera's locations. This setup has already been implemented (seen in the supplementary video); its evaluation is part of future work.

7.5 Summary

The *Vicarios* interface forms the basis of a software and control framework for intuitive remote robotic teleoperation for human-in-the-loop tasks. High-fidelity visualization and perception are key components of teleoperation, and an immersive interface provides inherent benefits in this regard. As stated earlier, the article builds on the recent approaches in research and adds the contributions of the integration of multiple features: viewpoint-independent motion mapping based on current point-of-view, real-time video and point-cloud from the remote environment to capture information (pose, dimensions, etc.) about dynamic elements, and the teleporting feature for intuitive and immersive execution. These features have shown to improve users' performance in tele-manipulation tasks. The questionnaire results demonstrate the improvement to the operator situational awareness that *Vicarios* offers, allowing consistent execution of remote teleoperation. Furthermore, we proposed an extension to the *Vicarios* framework to allow for an intuitive coupling between the user-driven teleporting in VR and a motion-capable remote camera, granting the possibility to overcome occlusions in the manipulation scene.

A note on ethical aspects of *Vicarios* - the key insight here is that the robotic platform in the remote environment is not autonomous, and is not interacting with other humans / living objects. Further, *Vicarios* does not require, nor does it gather personal information about the human operators. The system being based on the human-in-the-loop paradigm, the robot cannot act on its own, i.e.,, it cannot do something entirely different or detrimental, than what the human is commanding it to do. The improvement in situational awareness envisaged with *Vicarios*, is indeed for the human in-the-loop, and not for the robot. *Vicarios* is introduced here as a telerobotics interface in hazardous environments, where the human operator is always incharge. As noted in literature [31, 41], replacing human workers with robots in hazardous operations can lead to improvements in workplace safety. We believe that robots replacing humans is not always unethical, especially when contrasted with the goal of saving worker lives.

7.6 Limitations at the Current Stage

Although *Vicarios* contributes several new features that are helpful to the operator, there are some limitations to the framework at this stage, that we intend to address in our future works. For instance, *Vicarios* was not tested when rendering perception modalities other than vision (i.e.,., haptic and audio), to determine their impact on users'

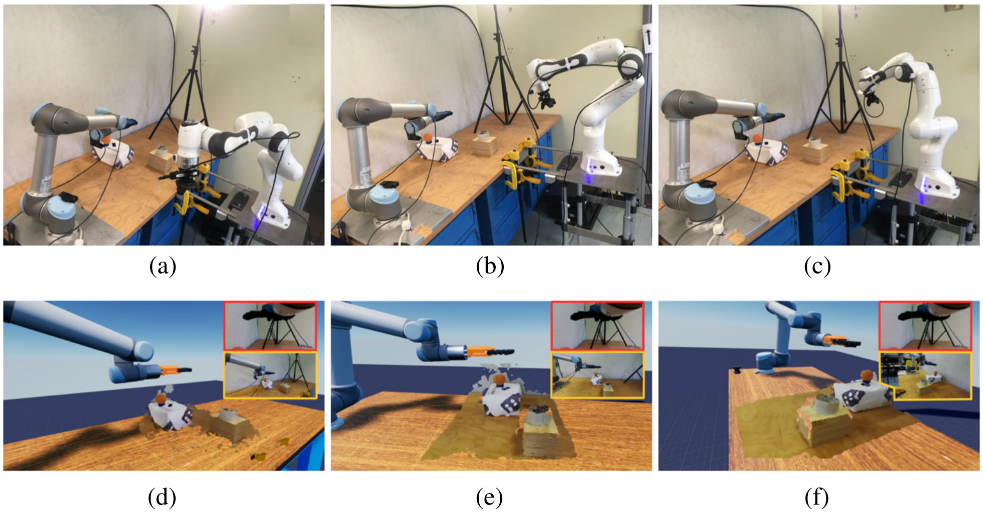


Fig. 8 *Vicarios* extension with the additional camera-robot (Franka Emika) arm to achieve changes in camera viewpoints to adapt to the teleporting feature. **a** Camera-robot pose 1, **b** Camera-robot pose 2, **c** Camera-robot pose 3 shows the hardware setup in the remote site. **d**, VR-teleport view 1 **e** VR-teleport view 2, **f** shows the actual user view within the interface. The red-bordered panel shows the view of the UR5-wrist-mounted camera, while the yellow-bordered panel shows the view from the camera-robot

performance [10]. This would open new perspectives in transmitting users' commands, such as voice or gesture control, as well as sensory substitution in some cases. Another limitation is that we have tested the system with only a few selected viewpoints while visualizing the virtual environment. Greater freedom in viewpoint selection can help as well as hinder performance [43]. Finally, latency is an inevitable problem in teleoperation interfaces that profoundly alters the operators' performance. We have included overall latency analysis in this article, and we intend to delve further to understand sub-component latency as well. Cloud-based solutions in telerobotics [16] can offer a promising alternative allowing complex computation to be offloaded to the cloud servers. The bandwidth and latency outcomes will need to be understood for such an approach.

8 Conclusions and Future Work

In this paper, the *Vicarios* VR interface was presented, forming the basis for an immersive and intuitive remote robotic teleoperation interface. It integrates: (i) the Unreal graphics engine for high quality VR rendering robotic platforms for manipulation in remote environments, viewed through the HTC system; (ii) real-time video and point-cloud streaming, with RGB-D cameras for real-time perception and remote environment update; (iii) user viewpoint changes using the teleporting feature in VR to overcome occlusions; and (iv) viewpoint-independent mapping between operator gestures and remote robot manipulator. All these features of *Vicarios* interface allow the user to freely explore the VR scene to better understand, view, locate, and interact with the remote environment with the aim of making it adaptable for demanding telerobotic domains, e.g., disaster response, nuclear decommissioning, telesurgery, etc.

As next steps, *Vicarios* shall be evaluated with respect to users' performance and learning, when allowing them to control the manipulator and the motion-capable camera during task execution. With dynamic unstructured remote environments, obtaining *a priori* information can be difficult, if not impossible. To address this, further evolution of the real-time capture and rendering of the remote environment in VR shall also be investigated. Finally, we will aim to implement and test the *Vicarios* platform when communicating via a wireless connection (e.g., 5G).

Acknowledgments We thank Patricia Yáñez Piqueras and Andrea Santangelo for their technical support in both human and robot experiements.

Author Contributions List of all authors: Abdeldjallil Naceri, Dario Mazzanti, Joao Bimbo, Yonas T. Tefera, Domenico Prattichizzo, Darwin G. Caldwell, Leonardo S. Mattos and Nikhil Deshpande.

All authors contributed to the study conception and design. Technical work was conducted by Abdeldjallil Naceri and Dario Mazzanti. Abdeldjallil Naceri and Nikhil Deshpande designed the user experimental study. Joao Bimbo contributed to robot arm (UR5) control. Yonas T. Tefera contributed to point cloud data streaming and delay measurements. User study, material preparation, data collection and analysis were performed by Abdeldjallil Naceri. The first draft of the manuscript was written by Abdeldjallil Naceri and Nikhil Deshpande. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding provided by Istituto Italiano di Tecnologia. This work was funded by INAIL (Istituto nazionale Assicurazione Infortuni sul Lavoro) and by the Lighthouse Initiative Geriatronics by StMWi Bayern (Project X, grant no.5140951), LongLeif GaPa gGmbH (Project Y, grant no. 5140953).

Declarations

Ethical Approval The testing and experimental procedures were approved by the "Ethics Committee of Liguria Region" (Italy), in accordance with the guidelines of the Declaration of Helsinki for research involving human participants

Consent to Participate Informed written consent was obtained from all participants involved in the study

Consent to Publish Individuals appeared in the current submission images/videos gave their consent for publication

References

1. Pisa/IIT SoftHand Software Package. https://github.com/CentroEPiaggio/pisa-iit-soft-hand. Accessed on 28-Feb-2019. [Online] Available: github.com/CentroEPiaggio/pisa-iit-soft-hand
2. RobCog-IAI: UE4 based Robot Simulator. https://github.com/robcog-iai/URoboSim. Accessed on 28-Feb-2019. [Online] Available: github.com/robcog-iai/URoboSim
3. Bejczy, A.K.: Virtual reality in robotics. In: 1996 IEEE Conference on emerging technologies and factory automation. ETFA '96, vol. 1, pp. 7–15 (1996)
4. Bicchi, A., Gabiccini, M., Santello, M.: Modelling natural and artificial hands with synergies. Philos. Trans. R. Soc. Lond. B Biol. Sci. **366**(1581), 3153–3161 (2011)
5. Bimbo, J., Pacchierotti, C., Aggravi, M., Tsagarakis, N., Prattichizzo, D.: Teleoperation in cluttered environments using wearable haptic feedback. In: 2017 IEEE/RSJ International conference on intelligent robots and systems (IROS), pp 3401–3408 (2017)

6. Bowman, D.A., McMahan, R.P.: Virtual reality: How much immersion is enough? Computer **40**(7), 36–43 (2007)
7. Buss, S.R.: Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods. Tech. rep., University of California, San Diego USA (2004)
8. Calzado, J., Lindsay, A., Chen, C., Samuels, G., Olszewska, J.I.: SAMI: Interactive, multi-sense robot architecture. In: 2018 IEEE 22^{nd} International Conference on Intelligent Engineering Systems (INES), pp. 317–322. https://doi.org/10.1109/INES.2018.8523933 (2018)
9. Catalano, M.G., Grioli, G., Farnioli, E., Serio, A., Piazza, C., Bicchi, A.: Adaptive synergies for the design and control of the Pisa/IIT SoftHand. Int J Robotics Res **33**(5), 768–782 (2014)
10. Chen, J.Y.C., Haas, E.C., Barnes, M.J.: Human performance issues and user interface design for teleoperated robots. IEEE Trans Sys Man Cybern Part C (Applications and Reviews) **37**(6), 1231–1245 (2007)
11. Christie, M., Olivier, P., Normand, J.M.: Camera control in computer graphics. Computer Graphics Forum **27**(8), 2197–2218 (2008). https://doi.org/10.1111/j.1467-8659.2008.01181.x
12. Christou, C.G., Aristidou, P.: Steering versus teleport locomotion for head mounted displays. In: De Paolis, L.T., Bourdot, P., Mongelli, A. (eds.) Augmented Reality, Virtual Reality, and Computer Graphics, pp. 431–446. Springer International Publishing, Cham (2017)
13. Crick, C., Jay, G., Osentoski, S., Pitzer, B., Jenkins, O.C.: Rosbridge: ROS for non-ROS users. In: Inproceedings of the 15^{th} International Symposium on Robotics Research (ISRR) (2011)
14. DeJong, B.P., Colgate, J.E., Peshkin, M.A.: Improving teleoperation: reducing mental rotations and translations. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, vol. 4, pp. 3708–3714 (2004)
15. Fiorini, S.R., Bermejo-Alonso, J., Gonçalves, P., de Freitas, E.P., Olivares-Alarcos, A., Olszewska, J.I., Prestes, E., Schlenoff, C., Ragavan, S.V., Redfield, S., Spencer, B., Li, H.: A suite of ontologies for robotics and automation [industrial activities]. IEEE Robotics & Automation Magazine **24**(1), 8–11 (2017). https://doi.org/10.1109/MRA.2016.2645444
16. Pignaton de Freitas, E., Olszewska, J.I., Carbonera, J.L., Fiorini, S.R., Khamis, A., Ragavan, S.V., Barreto, M.E., Prestes, E., Habib, M.K., Redfield, S., Chibani, A., Goncalves, P., Bermejo-Alonso, J., Sanz, R., Tosello, E., Olivares-Alarcos, A., Konzen, A.A., Quintas, J., Li, H.: Ontological concepts for information sharing in cloud robotics. Journal of Ambient Intelligence and Humanized Computing. https://doi.org/10.1007/s12652-020-02150-4 (2020)
17. Galvane, Q.: Automatic cinematography and editing in virtual environments. Ph.D. thesis, Grenoble University (2015)
18. Han, P.H., Hsieh, C.E., Chen, Y.S., Hsiao, J.C., Lee, K.C., Ko, S.F., Chen, K.W., Chou, C.H., Hung, Y.P.: Aoes: Enhancing teleportation experience in immersive environment with mid-air haptics. In: ACM SIGGRAPH 2017 Emerging Technologies, SIGGRAPH '17. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3084822.3084823 (2017)
19. Kaknjo, A., Rao, M., Omerdic, E., Robinson, L., Toal, D., Newe, T.: Real-time video latency measurement between a robot and its remote control station: Causes and mitigation. Wirel. Commun. Mob. Comput. **2018**(8638019), 19 (2018)
20. Kamezaki, M., Yang, J., Iwata, H., Sugano, S.: A basic framework of virtual reality simulator for advancing disaster response work using teleoperated work machines. Journal of Robotics and Mechatronics **26**(4), 486–495 (2014). https://doi.org/10.20965/jrm.2014.p0486
21. Kot, T., Novák, P.: Application of virtual reality in teleoperation of the military mobile robotic system taros. Int J Adv Robotic Sys **15**(1), 1729881417751,545 (2018). https://doi.org/10.1177/1729881417751545
22. Krijn, M., Emmelkamp, P.M., Biemond, R., de Wilde de Ligny, C., Schuemie, M.J., van der Mast, C.A.: Treatment of acrophobia in virtual reality: The role of immersion and presence. Behaviour Research and Therapy **42**(2), 229–239 (2004). https://doi.org/10.1016/S0005-7967(03)00139-6. http://www.sciencedirect.com/science/article/pii/S0005796703001396
23. Krotkov, E., Hackett, D., Jackel, L., Perschbacher, M., Pippine, J., Strauss, J., Pratt, G., Orlowski, C.: The DARPA robotics challenge finals: Results and perspectives, pp. 1–26. Springer International Publishing (2018)
24. Krupke, D., Starke, S., Einig, L., Zhang, J., Steinicke, F.: Prototyping of immersive HRI scenarios. In: Proc. 20^{th} Intl. Conf. on CLAWAR 2017, pp. 537–544 (2017)
25. Lipton, J.I., Fay, A.J., Rus, D.: Baxter's Homunculus: Virtual reality spaces for teleoperation in manufacturing. IEEE Robotics and Automation Letters **3**(1), 179–186 (2018)
26. Loup, G., Loup-Escande, E.: Effects of travel modes on performances and user comfort: A comparison between ArmSwinger and teleporting. International Journal of Human-Computer Interaction **35**(14), 1270–1278 (2019). https://doi.org/10.1080/10447318.2018.1519164
27. Mekuria, R., Blom, K., Cesar, P.: Design, implementation, and evaluation of a point-cloud codec for tele-immersive video. IEEE Trans Circuits Sys Vid Technol **27**(4), 828–842 (2017)
28. Melchiorri, C.: Robot Teleoperation, pp. 1–14. Springer London, London (2013)
29. Miguel, B., Andrew, S., Brian, C., Trey, S., Rodrigo, V.: HTC Vive: Analysis and Accuracy Improvement. In: IEEE/RSJ IROS 2018, pp. 2610–2615 (2018)
30. Milgram, P., Ballantyne, J.: Real world teleoperation via virtual environment modeling. In: Intl. Conf. on Artificial Reality and Tele-existence (ICAT'97) (1997)
31. Murashov, V., Hearl, F., Howard, J.: Working safely with robot workers: Recommendations for the new workplace. J. Occup. Environ. Hyg. **13**(2), D61–D71 (2016)
32. Murphy, R.R., Tadokoro, S., Kleiner, A.: Disaster robotics, pp. 1577–1604. Springer International Publishing. (2016)
33. Naceri, A., Chellali, R., Hoinville, T.: Depth perception within peripersonal space using head-mounted display. Presence: Teleoperators and Virtual Environments 20(3). https://doi.org/10.1162/PRES_a_00048 (2011)
34. Naceri, A., Mazzanti, D., Bimbo, J., Prattichizzo, D., Caldwell, D.G., Mattos, L.S., Deshpande, N.: Towards a virtual reality interface for remote robotic teleoperation. In: 2019 19th International Conference on Advanced Robotics (ICAR), pp. 284–289. https://doi.org/10.1109/ICAR46387.2019.8981649 (2019)
35. Naceri, A., Moscatelli, A., Chellali, R.: Depth discrimination of constant angular size stimuli in action space: Role of accommodation and convergence cues Frontiers in Human Neuroscience 9(September). https://doi.org/10.3389/fnhum.2015.00511 (2015)
36. Nowak, K.L., Biocca, F.: The effect of the agency and anthropomorphism on users' sense of telepresence, copresence, and social presence in virtual environments. Presence **12**(5), 481–494 (2003)
37. Olszewska, J.I., Barreto, M., Bermejo-Alonso, J., Carbonera, J., Chibani, A., Fiorini, S., Gonçalves, P., Habib, M., Khamis, A., Olivares-Alarcos, A., de Freitas, E.P., Prestes, E., Ragavan, S.V., Redfield, S., Sanz, R., Spencer, B., Li, H.: Ontology for autonomous robotics. In: 2017 26^{th} IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 189–194. https://doi.org/10.1109/ROMAN.2017.8172300 (2017)

38. Ouramdane, N., Otmane, S., Mallem, M.: Interaction 3D en Réalité Virtuelle - Etat de l'art. Technique et Science Informatiques **28**(8), 1017–1049 (2009). https://doi.org/10.3166/tsi.28.1017-1049. https://hal.archives-ouvertes.fr/hal-00376458
39. Park, S., Oh, Y., Hong, D.: Disaster response and recovery from the perspective of robotics. https://doi.org/10.1007/s12541-017-0175-4 (2017)
40. Peppoloni, L., Brizzi, F., Avizzano, C.A., Ruffaldi, E.: Immersive ROS-integrated Framework for Robot Teleoperation. In: 2015 IEEE Symposium on 3D User Interfaces (3DUI), pp. 177–178 (2015)
41. Pham, Q.C., Madhavan, R., Righetti, L., Smart, W., Chatila, R.: The impact of robotics and automation on working conditions and employment [Ethical, Legal, and Societal Issues]. IEEE Robotics & Automation Magazine **25**(2), 126–128 (2018). https://doi.org/10.1109/MRA.2018.2822058
42. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source Robot Operating System. In: IEEE ICRA Workshop on Open Source Software. IEEE (2009)
43. Rahnamaei, S., Sirouspour, S.: Automatic viewpoint planning in teleoperation of a mobile robot. J Intell Robotic Sys **76**, 443–460 (2014)
44. Rakita, D., Mutlu, B., Gleicher, M.: An autonomous dynamic camera method for effective remote teleoperation. ACM/IEEE International Conference on Human-Robot Interaction, pp 325–333. https://doi.org/10.1145/3171221.3171279 (2018)
45. Sanchez-Vives, M.V., Slater, M.: From presence to consciousness through virtual reality. Nat Rev Neurosci **6**(4), 332–339 (2005)
46. Sheridan, T.B.: Musings on telepresence and virtual presence. Presence: Teleoperators and Virtual Environments **1**(1), 120–126 (1992). https://doi.org/10.1162/pres.1992.1.1.120
47. Slater, M., Linakis, V., Usoh, M., Kooper, R., Street, G.: Immersion, presence, and performance in virtual environments: An experiment with tri-dimensional chess. In: ACM Virtual Reality Software and Technology (VRST), pp. 163–172 (1996)
48. Stotko, P., Krumpen, S., Schwarz, M., Lenz, C., Behnke, S., Klein, R., Weinmann, M.: A VR System for Immersive Teleoperation and Live Exploration with a Mobile Robot (November), pp. 3630–3637. https://doi.org/10.1109/iros40897.2019.8968598 (2020)
49. Theofanidis, M., Sayed, S.I., Lioulemes, A., Makedon, F.: VARM: Using Virtual Reality to Program Robotic Manipulators. In: Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA 2017, pp. 215–221 (2017)
50. Whitney, D., Rosen, E., Ullman, D., Phillips, E., Tellex, S.: ROS reality: A virtual reality framework using consumer-grade hardware for ROS-enabled robots. In: Proc. IEEE IROS 2018, pp. 5018–5025 (2018)
51. Witmer, B.G., Singer, M.J.: Measuring presence in virtual environments: A presence questionnaire. Presence **7**(3), 225–240 (1998)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Abdeldjallil Naceri (B.Sc. 2003, M.Sc. 2005, Ph.D. 2011) is Senior Research Scientist at Munich School of Robotics and Machine Intelligence (MSRM), Technical University of Munich (TUM). He is leading the research topic on Service Robotics and Human-Robot Interaction (HRI): Cognition, Perception and Interaction. His research expertises are human and robot hands, grasping and manipulation, impedance learning during HRI, perception and action in virtual reality (VR) and VR 3D user interfaces for teleoperation.

Dario Mazzanti is a technician at the Advanced Robotics Department of Istituto Italiano di Tecnologia. He has a PhD in Interaction Technologies, and a MSc in Computer Science Engineering. His main research interests are Interactive Environments, Virtual and Augmented Reality and Human Computer Interaction.

Joao Bimbo is a postdoc at the Mechanical Engineering and Materials Science Department of Yale University. He has a PhD in Robotics from King's College London and a MSc. in Electrical Engineering from the University of Coimbra. His research interests include robot grasping, tactile sensing, and physical human-robot interaction.

Yonas T. Tefera holds a BSc degree in Electrical and Computer Engineering, and a Master in Information and Communication Engineering from University Of Trento. Currently, he is a PhD student at Istituto Italiano di Tecnologia (IIT) and University of Verona (UNIVR), Italy. His research interests include virtual reality, gaze contingent rendering, point cloud and mesh compression and Dynamic 3D reconstruction.

Domenico Prattichizzo is with the University of Siena as full Professor since 2015. From 2002 to 2015 Associate Professor of Robotics at the University of Siena. Since 2009 Scientific Consultant at Istituto Italiano di Tecnoloogia, Genova Italy. In 1994, Visiting Scientist at the MIT AI Lab. M.S. degree in Electronics Engineering and the Ph.D. degree in Robotics and Automation from the University of Pisa in 1991 and 1995, respectively.

Darwin G. Caldwell is Founding Director of the Italian Institute of Technology (IIT), where he is also the Director of the Department of Advanced Robotics. He is or has been an Honorary Professor at the Universities of Manchester, Sheffield, Bangor, and King's College London in the UK and Tianjin University and Shenzhen Academy of Aerospace Technology in China. He is a Fellow of the Royal Academy of Engineering.

Leonardo S. Mattos (B.Sc. 1998, M.Sc. 2003, Ph.D. 2007) is a Permanent Researcher and Head of the Biomedical Robotics Laboratory at IIT. His research background includes medical robotics, teleoperation, computer vision, user interfaces, and augmented reality systems. Leonardo currently leads a team of 15 researchers and technicians at IIT. He was the PI and coordinator of the European project RALP - Micro-Technologies and Systems for Robot-Assisted Laser Phonomicrosurgery, and is currently leading the translational project Robotic Microsurgery, which is dedicated to the creation of new technologies for the detection and treatment of diseases in delicate organs.

Nikhil Deshpande is currently a Researcher and Team Leader at the IIT, Italy. His research interests include improving situational awareness for advanced robotic teleoperation systems, utilizing holistic approaches in VR/AR/MR. Previously, he focused on the design, development, and evaluation of novel surgeon-machine teleoperation interfaces, surgical devices, and controllers for robot-assisted surgeries. Nikhil received his Ph.D. in Electrical Engineering from North Carolina State University (NCSU) at the Center for Robotics and Intelligent Machines (CRIM) lab, in 2012, conducting his research in intelligent autonomous navigation, leveraging the computation in multiple, and small wireless sensor networks.