WFH-VR: Teleoperating a Robot Arm to set a Dining Table across the Globe via Virtual Reality

Lai Sum Yim[1], Quang TN Vo[2], Ching-I Huang[1], Chi-Ruei Wang[1], Wren McQueary[2]
Hsueh-Cheng Wang[1], Haikun Huang[2], and Lap-Fai Yu[2]

***Abstract*— This paper presents an easy-to-deploy, virtual reality-based teleoperation system for controlling a robot arm. The proposed system is based on a consumer-grade virtual reality device (Oculus Quest 2) with a low-cost robot arm (a LoCoBot) to allow easy replication and set up. The proposed Work-from-Home Virtual Reality (WFH-VR) system allows the user to feel an intimate connection with the real remote robot arm. Virtual representations of the robot and objects to be manipulated in the real-world are presented in VR by streaming data pertaining to orientation and poses. The user studies suggest that 1) the proposed telerobotic system is effective under conditions both with and without network latency, whereas a method that simply streams video does not. This design enables the system implemented at an arbitrary distance from the actual work site. 2) The proposed system allows novices to perform manipulation tasks requiring higher dexterity than traditional keyboard controls can support, such as setting tableware. All results, hardware settings, and questionnaire feedback can be obtained at https://arg-nctu.github.io/projects/vr-robot-arm.html.**

I. INTRODUCTION

The COVID-19 epidemic has prompted many people to work remotely from home in order to avoid in-person exposure at the work site. Nonetheless, many tasks requiring specialized skills and experience cannot be autonomously executed by robots reliably under actual uncertainty. Teleoperation solutions are helpful in these situations [1], [2]; however, the conventional approach using a 2D interface can be very cumbersome [3], particularly when operators are required to manage their views of the scene and command robot actuators using a keyboard and/or mouse [4]. Recent advances in Virtual reality (VR) devices have made it far easier to work remotely by immersing operators in a higher-fidelity virtual environment. VR provides an interface that allows users to specify points and transformations in an intuitive manner, but any communication with a remote system involves a signal delay, particularly in systems that depend on sensors.

In the current study, we employed a consumer-grade VR device (Oculus Quest 2) in fabricating an user-friendly virtual reality-based teleoperation system for controlling a low-cost robot arm (LoCoBot). The mobile base is kept stationary for this study. LoCoBots are are commonly available and can be deployed at scale.

*L.S. Yim and Q.T. Vo contributed equally to this work.

[1]Department of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, Taiwan.

[2]Department of Computer Science, George Mason University, USA.

Corresponding author email: hchengwang@g2.nctu.edu.tw

The research was supported by Taiwan's National Science and Technology Council (grants 111-NU-E-A49-001-NU, 110-2221-E-A49-124, and 111R10093Y-2). This work was funded in part by Qualcomm through the Taiwan University Research Collaboration Project. This work was supported by an NSF CAREER award (award#: 1942531) and an NSF FTW-HTF-R grant (award#: 2128867).

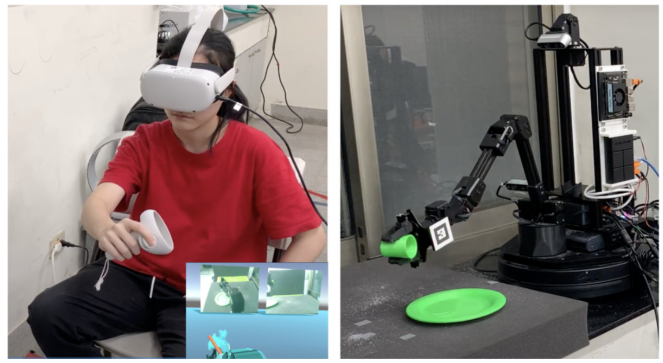


Fig. 1. A user controls a LoCoBot remotely to set up a dining table via virtual reality.

As shown in Fig. 2, the user obtains visual feedback through a VR headset. The user assigns the desired location for the end point of the robot simply by manipulating a VR controller. The VR interface includes a real-time image feed as well as its virtual counterpart, which is composed of 3D models. The two images are linked to the movement of the robot within the context of the surrounding area. Virtual counterparts are generated by estimating the pose of actual object via pose-estimation method. Thus, even when the actual objects are partially occluded in streaming image, the entire shape of the virtual counterpart can be seen in VR. This makes it easier for the user to operate the robot arm and interact with objects, while checking the state of the actual object via camera feeds. Our interface can display not only the actual robot state (opaque robot arm in Fig. 2) but also the commands of the end point via the user's VR controller (transparent virtual fixture in Fig. 2). Besides, pressing the button on the VR controller can deliver the commands to the actual robot. The two main movement functions (adjusting the opening of the gripper, and moving the end-effector) only track the user's motions when a specific button is pressed, in order to eliminate unintentional movements of the robot. Further, through transparent virtual fixture, the user can practice and pre-see what the robot arm will do before making the orders.

There has been significant attention and competition [5] with regard to service robots' grasping and manipulation tasks, such as setting up a dining table. Through virtual reality, our users controlled a robot arm to arrange tableware (e.g., plates, forks) to match target positions and to perform tasks (e.g., pouring water). We recorded the time for performing different tasks and the users' feedback about using our system. The functionality of the system was assessed in user studies involving participants with no previous experience performing such operations. The major contributions of this work include the following:

- **Effectiveness of visualization methods under reduced frame rate**. The proposed VR robot teleoperation system, in which 3D virtual counterparts are presented via a virtual interface, was found effective against decreased frame rate compared to video stream visualization methods. Different manipulation strategies adopted by different participants are analyzed.
- **Efficacy of VR for the group with/without previous demonstration vs. conventional keyboard control methods**. We evaluate the VR robot teleoperation system using different dining table tasks and comparing it with alternative approaches keyboard control via a 2D screen.
- **Evaluation of how long it takes to master the teleoperation for novice participants**. The efficacy of the proposed system was demonstrated in a user study involving novice participants.

II. RELATED WORK

A. Telerobotics

Teleoperation by a human operator is often the only practical alternative when dealing with grasping and manipulation tasks that are too specific for autonomous solutions [4]. Telerobotic systems implement commands and relay information back to the operator. Control architectures can be classified as (1) direct control, (2) supervisory control, and (3) shared control [6]. Direct control implies that all slave operations are controlled directly by the user via a master interface, such that the system does not require innate intelligence or the ability to operate autonomously. Supervisory control implies a sparse connection between the user and a largely autonomous telerobot. In these systems, the operator sends only high-level commands, and the telerobot refines the tasks autonomously. One approach to supervisory control involves telesensor programming [7], in which robot tasks comprise elementary moves representing different subtasks described by the initial and final states. The transition from one subtask to the next subtask (i.e. the recognition that the goal state of the elemental move has been reached) is performed heuristically. Shared control implies that in the execution of a task, the commands are shared by the operator (direct control) and the robot (local autonomy). Telerobotic systems that suffer from significant latency tend to benefit from sensor-based or model-based programming. Some researchers have investigated recreating the remote environment with stimulated time delay model. By using Augmented reality to allow a operator to teach and operate a robot arm to do manipulation tasks. [8]. Our user studies were carried out at a real network connection to examine the effects of the poor network connection on novice/expert participants.

B. VR Teleoperation

Intuitive interaction is the cornerstone of efficient task execution in teleoperations. However, most conventional teleoperation schemes rely on computer monitors and joysticks or keyboards to actuate the robot. These 2D interfaces are cumbersome and workload-intensive (i.e., they require significant mental effort) [3], [4]. Utilizing VR in robotic systems, high-fidelity graphic renderings and native changes in viewpoint can assist in planning robot motions and overcoming the limitations of 2D visual interfaces. In addition, it involves the direct mapping of VR hand controllers to robot manipulators via an interface [9], with having 3D spatial information can assist in immersive controlling robot. Such VR teleoperation system also can employ exoskeletons for the operator to teleoperate bimanual robotic avatar [10], [11], [12] or apply motion capturing systems to teleoperate a aerial manipulator [13] and a hyper-redundant robot [14]. Researchers have proposed mobile-robot teleoperations aimed at validating the functionality of immersive VR environments [15], [16], [17], [18], [19], [20]. In most previous studies, human following and motion planning are implemented separately, with the primary focus on the effects of latency on the synchronization and positioning of targets in the real world and those represented in the VR system. When performing tasks that involve dexterity, robots generally perform multiple pick-up-and-place tasks [21], [22], which means the user is not kept interacting with the actual object.

In dealing with the distance between the gripper and object, some researchers have employed a wrist camera by which to read a range sensor and project a stereoscopic view of the arm [9]. Images from the camera can also be linked to a hand controller to enable constant monitoring during manipulation [3]. Despite limited success in the performance of VR tasks, it should be noted that the streaming of camera images via VR remains a special application of 2D interfaces. An approach to build virtual counterparts is rendering colored 3D point cloud from a remote depth camera mounted on the wrist [18], [21], [23], [24], [25], [26]. As transferring large amounts of point-cloud data may lead to a burden over network, some researchers proposed an automated object detection and streamlined data transfer method. The system executes an object detection and segmentation algorithm. The raw point cloud data is then replaced by virtual objects to reduce the amount of transferred data [21], [22]. However, the target object is often occluded by the robot arm, resulting in incomplete point-cloud data and failure in reconstructing the actual object. Also, it is possible to build virtual counterparts by fiducial system for the localization and tracking of objects relative to the robot itself [3]. Nonetheless, affixing tags to objects is difficult in many service robot applications. In the current study, we employed DOPE [27], which applied deep neural networks for 3D object detection and pose estimation in our VR interface. Using DOPE, our approach only needs an RGB image to do the pose estimation.

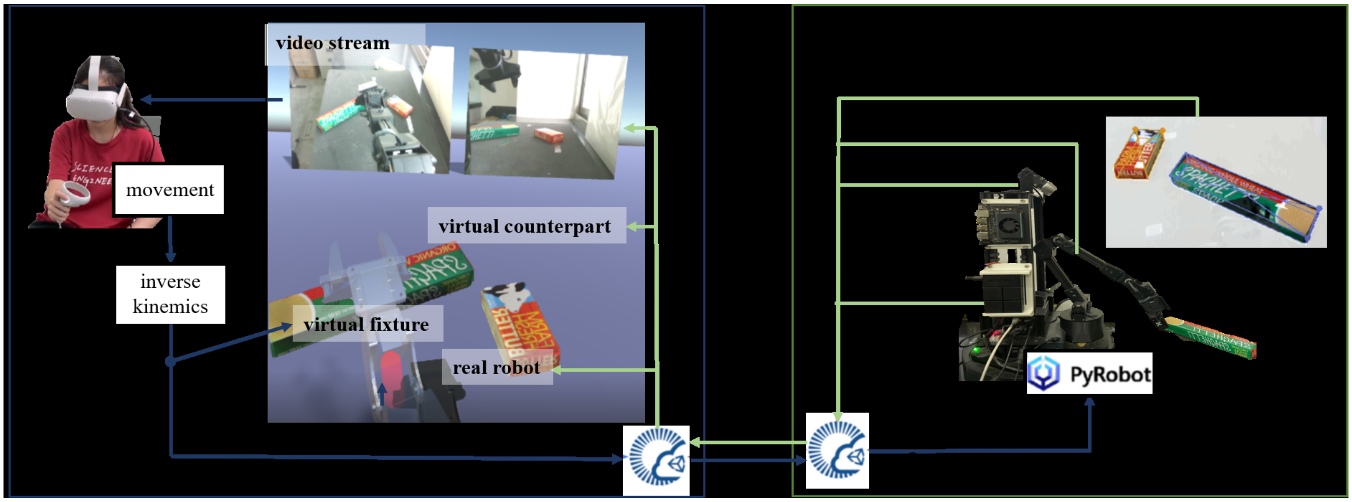


Fig. 2. System architecture of VR-based teleoperation system, which can be operated over the internet from anywhere in the world. In addition to the two real-time image feeds, virtual representations reflect the means by which users naturally interact with a target. Translucent virtual representations are used to indicate the predicted state of the robot after it receives a command.

III. PLATFORM DESCRIPTION

Figure 2 presents an overview of the proposed approach in which the user and actual robot are separated by an arbitrary distance. We employed Photon Unity Network 2 (PUN2) [28], which is a real-time cloud framework used to host online multiplayer games for Unity developers, for communication tasks over the internet and an Oculus Quest 2 VR headset to visualize the virtual environment while manipulating the robot arm. Within the virtual environment, actual objects are replicated via 3D reconstruction or CAD modeling. In the following experiment, we also leverage YCB object and model set [29], which is a benchmark in robotic manipulation research. The actual robot arm and gripper are controlled by guiding an identical virtual robot arm, which is modeled by the open-source LoCoBot's URDF model by Meta Research [30] (i.e., a robot arm model with the same joint configuration and dimensions as the actual robot arm).

ROS# [31], an open-source software library for Unity to communicate with ROS via .NET applications, and Rosbridge [32], an open-source software library providing a JSON API to access ROS functions for non-ROS programs, are used to build a WebSocket, which allows two-way communication between ROS and Unity data transfer. The remote environment data, including the estimated actual object pose, LoCoBot state and the streaming image (optional), is sent to Unity. Virtual objects are synchronized with actual objects by continuously performing pose estimation using Deep Object Pose Estimation (DOPE) [27] to process RGB images captured using a camera attached to the base of the actual robot, shown in Fig. 3. On the other hand, the control command from the VR user, including the desired joint state and gripper state for the actual robot arm, is sent to the computing unit in remote site. The proposed system enables users to observe virtual scenes that are synchronized with corresponding actual objects in the physical workplace, and to remotely control the robot arm. Even with the least information capable to transmit (robot state, object poses), the manipulation task can be achieved. It also gains the advantage of minimizing latency to make manipulation more intuitive. It will be discussed in section IV. The current study on VR interface is investigated from the first-person point of view (FPS) of the VR User.

A. Hardware

The proposed teleoperation platform was implemented using a consumer-grade VR device (Oculus Quest 2; $420) and a low-cost robot (LoCoBot; $5,000). The Oculus Quest 2 system provides a head-mounted display comprising a singular fast switching LCD panel with a resolution of 1832 $\times$ 1920 per eye and a refresh rate of 120 Hz. It is also equipped with two Oculus Touch hand controllers with 6 DoF pose tracking using infrared LEDs, thereby allowing comprehensive tracking in a 3D space by the Oculus Quest 2 constellation system. Fig. 3 presents the LoCoBot robot used in this study with a control system comprising the following components: an Intel RealSense RGB-D Camera D435, a Jetson Xavier NX, a WidowX 200 Mobile Arm (5 DOF), an Intel NUC, and a Kobuki Base. In addition to the original camera at the top of the robot, we attached an Intel RealSense RGB-D Camera D435 near the base of the arm to capture RGB images for object pose estimation. Note that we opted not to use the original camera due to the likelihood of robot arm occlusions, which could cause pose estimation failures. The teleoperation system was developed in Unity, a 3D game engine that supports major VR headsets, including the Oculus Quest 2.

B. Pose Estimation

DOPE [27] is used to estimate object poses on the robot side in order to update the poses of virtual objects on the operator side. As shown in Fig. 3, the top camera was used for real-time image streaming, whereas the lateral camera was used to capture images with which to estimate objects' poses. The robot was also equipped with Xavier NX which

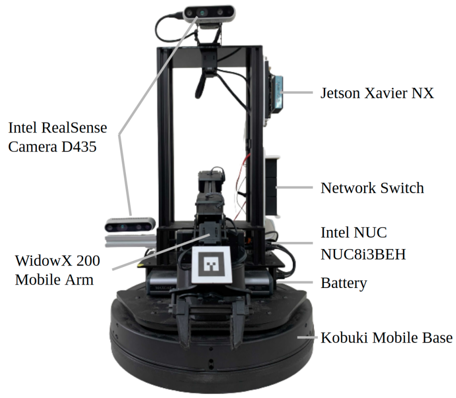


Fig. 3. The proposed robot setup.

for performing DOPE calculations. We employed the popular YCB [29] object model as a reference to facilitate the rendering of objects (e.g., a Domino sugar box). In case of failure pose-estimation by DOPE, a physical engine in Unity simulates the dynamics of the virtual objects with the aim of preserving interactions until the pose is updated using data from the actual object. The representation of virtual objects changed from opaque to transparent.

C. Robot Arm Control in Virtual Reality

1) Visual Interface: Effective control over the robot in virtual reality requires the user to have a clear spatial understanding of the environment to be manipulated on the remote side. In our system, five visual elements are present in the virtual scene. **(a)Apaque virtual robot** is for synchronization with the actual robot while **(b)transparent virtual robot** is for synchronization with the user's movement, which are mapped by the Oculus Quest right controller. Thereby, allowing the user to move their arm freely, while guiding the movement of the virtual robot arm with the action visualized in real time. Beside, **(c)apaque virtual objects** are synchronized by DOPE [27]. Nonetheless, self-occlusion often occurs when the robot arm is picking up an object, in which case DOPE may fail to estimate the object's pose. To mitigate this problem, the object in this situation is rendered from apaque to **(d)transparent virtual objects**, to let the user know that the object's pose is uncertain. Further guidance is **(e)designated screen** in the virtual working environment, which streaming RGB video signal from the cameras in the remote environment. This is intended to clarify the positions of objects and events at the work site to the user.

2) Control Interface: To minimize communication latency and avoid severe computational expense on the remote side, we replaced the PyRobot inverse kinematic solver with Unity's kinematic chain and IK solver. The goal-point is set on the end effector of the virtual robot, which is mapped onto the right controller. Once the goal-point and constraints have been set, the FABRIK algorithm [33] is used to compute the state of each joint of the virtual robot. The user then confirms the desired trajectory by pressing the grip button on the right controller, whereupon the host's Unity program publishes the joint state of the virtual robot to the Rosbridge server [32] via ROS# [31]. This triggers a handler script to read it and uses PyRobot API [34] at the NUC to set the joints of the actual robot arm. By circumventing PyRobot IK, this approach avoids one step behind the awareness of the action, thereby making it possible to control the actual robot in real time when a movement is performed by the user.

D. Network Control

PUN2 [28] allow the VR user (client) control the robot (host) from any location that has access to the internet. We leverage the Remote Procedure Calls Protocol (RPC) of PUN2 to send and receive data remotely and synchronize the virtual scene with minimum latency. The VR user sends their controller inputs and desired joint state via RPC, for use in adjusting the robot's position. Accordingly, the video streams and estimated poses of actual object are sent from the host to the VR side for synchronizing the virtual working space with the actual remote environment. We found that there is no significant delay (i.e., no longer than 1 second) from the synchronization of the actual robot state to VR and the control command from the client to the host. However, video streaming and virtual counterparts suffer from frame rate reduction, from 14 fps and 1.4 fps accordingly dropping to 0.75 fps; this is largely due to the bandwidth limit imposed by PUN and internet connection speed. We investigated the user experience of teleoperating a robot arm to set a dining table. The two ends of the experiment were set in two universities located halfway around the world. Specifically, the VR side was set up at George Mason University in Virginia, US, and the actual robot was set up at National Yang Ming Chiao Tung University in Taiwan.

IV. Human-robot-interaction Experiments

The effectiveness of the proposed system in remote teleoperation tasks was evaluated by conducting a user study involving three experiments. All of the participants (ages 20-40) lacked any prior experience with our platform. Experiment 1 focused on the visualization methods used in the proposed system, including video (in the form of a 2D monitor interface) and the virtual counterpart (rendered via pose estimation). Experiment 2 focused on evaluating how well our proposed system allows even novices to perform tasks of high dexterity. In accordance with the Robotic Grasping and Manipulation Competition at IROS 2021 [35], we employed 5 tasks involving setting a table. The final experiment focused on mapping the skills developed in one task to another task. The setups of the experiments are shown in Fig. 4. In addition, the results of a 7-point Likert scale were used for subjective analysis while the completion time and score were used as an objective index of difficulty.

A. Experiment 1: Methods of Visualization

The first manipulation task involved pushing one object and then stacking another object on it. Both of the objects used in this experiment were selected from the YCB object

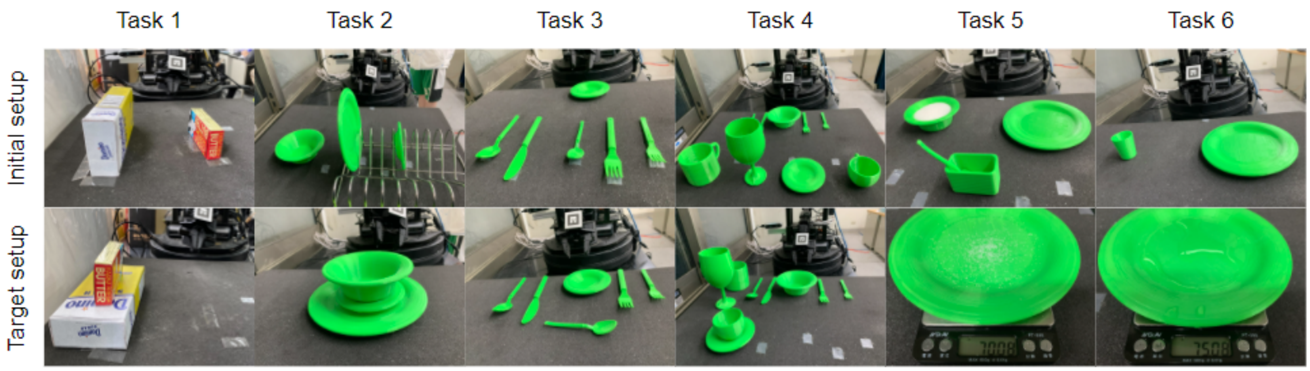


Fig. 4. We evaluated our system based on dining table setting tasks, including: (1) pushing and stacking of boxes; (2) picking up and stacking plates and bowls ; (3) picking up and placing multiple implements such as cutlery; (4) rearranging glasses and cups ; (5) smoothly scooping sugar grains from a set distance; (6) smoothly pouring water from a set distance. The proposed system allows even novices to perform tasks of high dexterity through the remote manipulation of a robot arm via the internet from anywhere in the world.

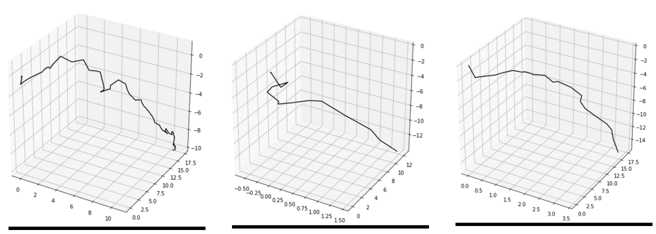
set [29]. The objects were visualized in VR using (a) one real-time image, (b) two real-time images from different perspectives, or (c) virtual counterparts rendered via pose estimation.

A local network was set up via a wired Ethernet cable connection between the computing unit of the LoCoBot and the desktop, which for running Unity program, in the remote environment, while the Oculus Quest 2 was connected via a USB cable to the desktop in the operator site. We first conducted a test under two different network modes: 1) Local mode (without PUN2) and 2) Global mode (with PUN2) by changing the gripper state and going through a list of designated goal points to estimate latency. The method here was for all computing units to do the time synchronization with the public time sever at the beginning. Then, when the button was triggered by the user, the current UTC date-time was recorded and a "trigger message" was passed though the internet to the remote side. Once the message was received, a "feedback message" was sent back to the operator side immediately, the time different between "trigger time" and "feedback time" on the operator site was considered as the communication time. We assume that it is the same as the update time of virtual robot arm from the actual robot arm. Based on the recorded video, we collected the update time from video streaming when the desired state of the virtual robot arm was updated. 10 transferring samples were collected for analysis. For Local mode, the test was performed by a VR user in the same building as the actual robot. The update time was approximately 0.1s after the communication time ($\sim$0.03s). By contrast, for Global mode, the test was performed by a VR user in the US while the robot remained in Taiwan. The update time was approximately 0.2s after the communication time ($\sim$0.35s). This indicated that less than 0.2s for Local mode and 0.6s for Global mode will work properly with our system. In addition to the latency discussed here, we will also discuss how decreased frame rate will affect the participants' performance.

To compare the various approaches to visualization, we recruited 12 participants who were novices in the use of VR systems (i.e., with 5 or fewer hours of experience). We

(1) Top camera (2) Lateral camera (3) Virtual counterpart

Fig. 5. **Initial views of different visualization methods in experiment 1.** Method(a), one-image stream(1), includes only the image top camera. Both objects were nearly occluded by the robot arm. Method(b), two-image stream(1+2), includes images from both the top camera and the lateral camera, providing the user with a more comprehensive view of the work space. Method(c), virtual counterparts(only 3), generates the whole object model in VR.



Method(a) - (1) Method(b) - (1+2) Method(c) - (only 3)

Fig. 6. **The performance of trajectory in experiment 1 by an expert**. (i.e., a reference point (gripper_link relative to the base_link of the LoCoBot) from the initial position until the first contact against the yellow box via all visualization methods. (unit: cm) The strategy for Method(a) overcoming the occlusion by moving the robot arms several times manifests in the jagged line. The trajectory for Method (b) became smoother because of with 2 different perspectives. Method(c) allowed the expert user to complete the task directly and quickly.

decreased the frame rate (originally 14 fps from the camera feed and 1.4 fps for the virtual counterparts) to 1 fps (for all methods) in order to simulate the frame loss would expect to encounter when using a poor network connection. Half of the participants manipulated the robot using the video stream first (a and b) and then using virtual counterparts (c). The other half of the participants performed this sequence of operations in the opposite order. Fig. 7 displays the average completion time, which was used to quantify the efficiency of the methods. In terms of completion time, the use of virtual counterparts proved most helpful. It minimized the degree of variance in the outcomes as well as the number of task failures, regardless of the frame rate, such that most

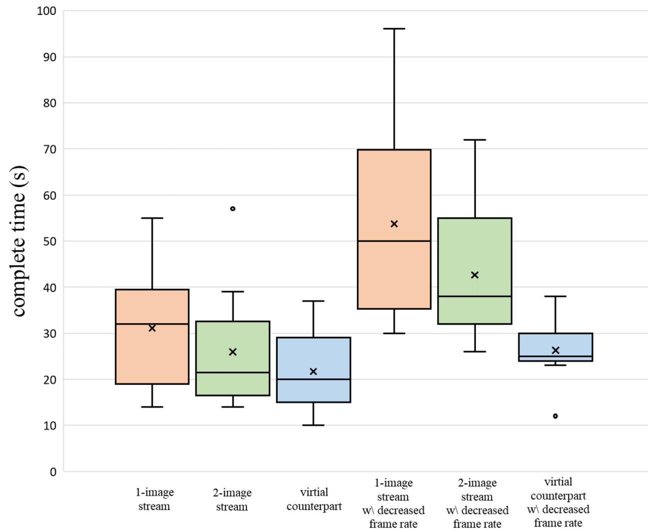


Fig. 7. **Completion times obtained via various visualization methods using normal frame rate and decreased frame rate by novice in experiment 1.**

of the participants were able to complete the task in their first try. Note that under a lower frame rate, the average time top completion was slightly longer (>5s) than under a normal frame rate. Even though the distance of the field was provided positional information like binocular vision in Method(b), the participants preferred to use the virtual counterpart, regardless of latency. When using Method(b) with a lower frame rate, the participants expressed that they felt uncoordinated. Taken together, these results reveal that a virtual model based on pose estimation facilitates robot manipulation, even when operated on a poor network. Some failure cases occur when the end-effector is approaching the box. Although it is assumed that the gripper is able to pick up the box with at least 38mm at a fully-opened state(50mm), the box was sometimes hit down by inappropriate approaching. The failure was mainly due to the error of pose-estimation by DOPE [27] or hand-eye camera calibration at the beginning. Fig. 5 shows the initial views of different visulization methods and Fig. 6 shows the performance of trajectory was drawn by an expert (>20 hours operating VR systems). As for the virtual counterpart from the lateral camera, it allowed the expert user to complete the task smoothly and quickly. This can be attributed to the fact that DOPE[27] was able to estimate the pose of actual object even partially occluded. Hence, the expert user was able to benefit for interactions with the object. Based on the above results, Method (c), virtual counterpart, was user-friendly and promising for both novices and expert users.

B. Experiment 2: VR vs. Keyboard Controls and Task Performance Benchmark by User Groups

The experiment tasks are inspired by the multi-year IROS Robotic Grasping and Manipulation Competition - Service Robot Track [35]. The objects we used in the experiments are also inspired by the YCB objects [29] and common objects on a dining table. Due to the hardware payload and gripper limitations, we chose to scale down the objects by modeling CAD models followed by 3D printing. Such modifications also considered easy replication of our work using LoCoBot hardware. All CAD models will be publicly available. There were other tasks designed in the competition, but we found them infeasible for the LoCoBot setup, such as the ice cubes task (which requires touch sensing) and the sugar packet task (which requires dual arms). Figure 4 depicts the five tasks (2-6). We chose a range of tasks and modified them to fit with our robot setup.

TABLE I

WE RECRUITED A TOTAL NUMBER OF 28 PARTICIPANTS TO CONDUCT THE 5 TASKS. THE BENCHMARK RESULTS OF AVERAGE SCORE AND TIME SPENT IN EACH TASK IN EXPERIMENT 2 ARE SHOWN.

N of Participants	KM-Expert 1	VR-Novice-B 10	VR-Novice-A 18	VR-Expert 1
Completion Time (unit: s)				
1: plates/bowl	197.3	139.6	133	102.8
2: tablewares	143.5	135.5	135.7	92.7
3: glasses/cups	260.8	169.3	161.4	110.8
4: sugar	86.3	156.1	58*	50.5
5: liquid	57.8	69.4	44.4	47
Score (unit: pts)				
1: (120) plates/bowl	112	100	108.89	110
2: (300) tablewares	276	271	255.56	282
3: (70) glasses/cups	66	66	67.5	70
4: (50) sugar	25	37.5	44.44*	50
5: (50) liquid	25	30	41.67	50

* The participants in this group only had one chance to transport sugar grains, whereas the other groups could attempt the task repeatedly until they wished to stop.

We recruited 28 novices to use our VR teleoperation system for the 5 tasks feasible for our hardware setup. All participants were first-time users. After some instruction, the participants wore the VR head mount and used the hand controller to interact with the objects. They were not instructed to grasp the objects in a specific way, and therefore all possible motion primitives, such as pushing, grasping, and placing, were allowed. In Tasks 2 to 5, there was a 5-minute time limit for each task, but there was no limit in terms of number of attempts made within that time constraint. In Task 6, users were only allowed one attempt to transport the fully-filled water cup. We also reported a baseline using keyboard and monitor (KM) inputs by a researcher experienced with our VR teleoperation system, to perform 5 trials of each task. The KM teleoperation method is provided by Pyrobot API [34]. The position of the end-effector is controlled by 'w,s,a,d,z,x' keys and directly adjusting two last joints of the robot arm to achieve like 'roll-pitch' function of the end-efftector by 'h,j,k,l' keys and one keystroke corresponds to 10 mm or ∼6 deg. We evaluated performance by counting the scores following the rules in the 2021 competition [35].

Group A (18 novices) were given some tips and allowed to watch previous participants before starting each task. Each participant in group A performed 2-3 trials with different tasks. Group B (10 novices) was a control group that did not receive any instruction. Each participant in group B performed 1 trial for each task. In total, 50 trials and 44 trials evenly distributed for each task were collected. Table I shows the average scores and completion time of using a VR video stream and controller (VR-VS) versus keyboard and

TABLE II

THE COMPETITION TIME, SCORE AND P-VALUE BETWEEN GROUPA AND GROUP B IN EACH EXAM FOR EXPERIMENT 3.

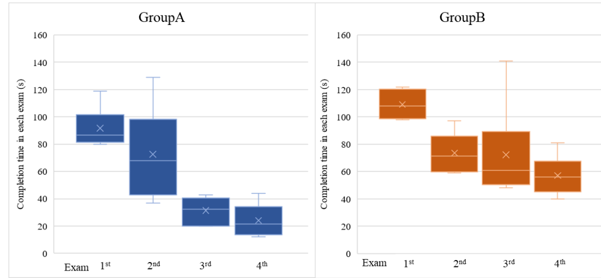
	Competition time (s)			Score		
	p-val	mean(A)	mean(B)	p-val	mean(A)	mean(B)
E1	0.358	96.7	109.2	**0.009**	**41.7**	**8.3**
E2	0.474	72.5	73.5	0.383	37.5	33.3
E3	**0.010**	**31.3**	**72.3**	0.500	45.5	45.8
E4	**0.001**	**24.0**	**57.2**	0.500	41.7	41.7

*p-val: p-value
mean(A): the mean value among GroupA
mean(B): the mean value among GroupB

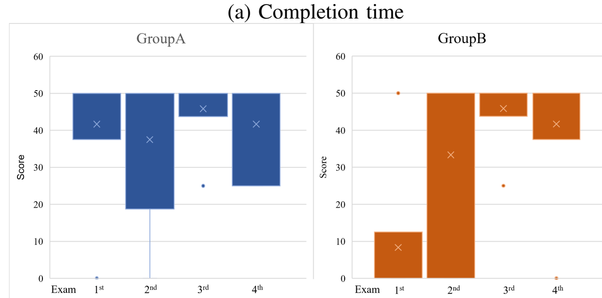
monitor (KM). Overall, both VR-VS and KM methods were able to complete all tasks. However, the execution time for VR-VS was shorter than that by KM. We observed that the VR interface allows different **motion primitives** like pushing or fine-tuned adjustment of the positions or orientations of the knife and fork. This affords the user multiple strategies to fine-tune incorrect motions, such as a poorly aligned pick-up-and-place maneuver. VR showed superior performance in Task 5 and Task 6 than KM by better **fluency in 6 DoF grasping**. During transport, the sugar grains or water tended to drop or splash out when controlled by KM due to a sequence of noncontinuous strokes. Using VR led to smoother and more intuitive control than KM. VR was also useful in Task 5; as the spoon reached into the sugar grain container, the scooping required **dexterous manipulation** to successfully retrieve the desired amount of sugar grains. Such skills were difficult to perform using the KM method. Note that the sequence of actions to perform the Task 6 appears simple, but is in fact far more difficult than the other tasks for the novices. For example, if the cup is not tilted smoothly across the critical point to pour out the water, then most of the water falls diagonally (hanging onto the cup's outer wall) rather than falling vertically. However, our results do not provide evidence that an opening expert demonstration improved participants' performance on 5 tasks. From the background investigation, the participants in group B were all inexperienced with VR but experienced with robotics, whereas the participants in group A, 15 having VR experience and 4 having robotics experience. We discovered that the participants with VR experience, who are familiar with hand controllers, perform dexterous manipulation more proficiently than Group B. Nevertheless, we find that our proposed system allows even novices to perform tasks of high dexterity. We further explore how the novice to become expert in the next experiment.

C. Experiment 3: Practice Makes Perfect

We conjecture that practice will make perfect, but how long does it take, and what kinds of practice could best allow a novice to master teleoperation? An experiment involving both practice and exams was set up to investigate. Task 6, pouring water on a plate, was selected for this experiment because previous experiments observed that tasks of this type require dexterous skills for teleportation. We grouped 12 novices into GroupA(with practice) and GroupB(without practice). This task, taken as the exam, was executed 4 times



(a) Completion time



(b) Score

Fig. 8. **The completion times and the scores of Group A (with practice) and Group B (without practice) in each exam for experiment 3.**

for each participant. The main difference was that GroupA did the practice before every exam. Two practice actions would be applied, (1) operate the gripper to approach a yellow box (the same box as in Task 1) 3 times, but not pick it up, for 3 minutes, and (2) draw the trajectory in the air with the box in hand for 1 minute. On the other hand. Group B simply rested for 4 minutes. It should be mentioned that the motion primitives during the practice were different from the exam. The results are displayed in Fig. 8 and Table. II. The performances of all participants had significantly improved. They completed the task in a shorter time and got higher scores. On the 1st exam, the p-value of score (<0.05) indicated that there is statistically significant difference between two groups and we deduced that GroupA has significantly better manipulating skills than GroupB due to practices. On the 4th exam, all participants in GroupA completed the task in less than 45s, the performance is very similar to the expert user (44s). Almost all participants in GroupA highly agreed that they felt more confident in the exam after doing several practices (average: 6.5/7), and found that practice was helpful for mapping the skills from practice to the exams (average: 6.5/7). As for GroupB, although the average scores in the last 2 exams were similar to GroupA, the average completion times were much higher (>30s). The reason here could be that GroupA has practiced and felt more confident for the exams; hence, they tended to spend less time just to get a good-enough score. In summary, it took a predictable time to make a novice have an expected performance. Furthermore, performing similar motion primitives was helpful for dexterous manipulation.

V. CONCLUSION

Our paper presents a novel approach to the VR teleoperation of a robot arm. Our toolkit will be released for free to facilitate adoption and future extension. Current limitations and possible extensions include the following: a) pose estimation failures due to self-occlusion, which could be addressed through the use of additional cameras; b) limited flexibility due to the use of only one robot arm with a gripper hand, which could be addressed by using an additional arm and a five-finger hand for performing more sophisticated manipulations; and c) failure to consider deformable objects (e.g., clothes), which may be addressed by estimating the 3D geometry of objects in real time.

References

[1] Z. Li, P. Moran, Q. Dong, R. J. Shaw, and K. Hauser, "Development of a tele-nursing mobile manipulator for remote care-giving in quarantine areas," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3581–3586.

[2] J. Li, Z. Li, and K. Hauser, "A study of bidirectionally telepresent tele-action during robot-mediated handover," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2890–2896.

[3] C. Barentine, A. McNay, R. Pfaffenbichler, A. Smith, E. Rosen, and E. Phillips, "A vr teleoperation suite with manipulation assist," in *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 2021, pp. 442–446.

[4] D. Whitney, E. Rosen, E. Phillips, G. Konidaris, and S. Tellex, "Comparing robot grasping teleoperation across desktop and virtual reality with ros reality," in *Robotics Research*. Springer, 2020, pp. 335–350.

[5] Z. Liu, W. Liu, Y. Qin, F. Xiang, S. Xin, M. A. Roa, B. Calli, H. Su, Y. Sun, and P. Tan, "Ocrtoc: A cloud-based competition and benchmark for robotic grasping and manipulation," *arXiv preprint arXiv:2104.11446*, 2021.

[6] G. Niemeyer, C. Preusche, S. Stramigioli, and D. Lee, "Telerobotics," in *Springer handbook of robotics*. Springer, 2016, pp. 1085–1108.

[7] B. Brunner, G. Hirzinger, K. Landzettel, and J. Heindl, "Multisensory shared autonomy and tele-sensor-programming-key issues in the space robot technology experiment rotex," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, vol. 3. IEEE, 1993, pp. 2123–2139.

[8] H. Beik-Mohammadi, M. Kerzel, B. Pleintinger, T. Hulin, P. Reisich, A. Schmidt, A. Pereira, S. Wermter, and N. Y. Lii, "Model mediated teleoperation with a hand-arm exoskeleton in long time delays using reinforcement learning," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020, pp. 713–720.

[9] 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, no. 1, pp. 179–186, 2017.

[10] C. Lenz and S. Behnke, "Bimanual telemanipulation with force and haptic feedback and predictive limit avoidance," *CoRR*, vol. abs/2109.13382, 2021.

[11] M. Schwarz, C. Lenz, A. Rochow, M. Schreiber, and S. Behnke, "Nimbro avatar: Interactive immersive telepresence with force-feedback telemanipulation," *CoRR*, vol. abs/2109.13772, 2021. [Online]. Available: https://arxiv.org/abs/2109.13772

[12] C. Zhou, L. Zhao, H. Wang, L. Chen, and Y. Zheng, "A bilateral dual-arm teleoperation robot system with a unified control architecture," in *2021 30th IEEE International Conference on Robot Human Interactive Communication (RO-MAN)*, 2021, pp. 495–502.

[13] G. A. Yashin, D. Trinitatova, R. T. Agishev, R. Ibrahimov, and D. Tsetserukou, "Aerovr: Virtual reality-based teleoperation with tactile feedback for aerial manipulation," *CoRR*, vol. abs/1910.11604, 2019.

[14] A. Martín-Barrio, J. J. Roldán-Gómez, I. Rodríguez, J. del Cerro, and A. Barrientos, "Design of a hyper-redundant robot and teleoperation using mixed reality for inspection tasks," *Sensors*, vol. 20, no. 8, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/8/2181

[15] Y. Mizuchi and T. Inamura, "Cloud-based multimodal human-robot interaction simulator utilizing ros and unity frameworks," in *2017 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2017, pp. 948–955.

[16] F. Okura, Y. Ueda, T. Sato, and N. Yokoya, "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.

[17] P. Stotko, S. Krumpen, M. Schwarz, C. Lenz, S. Behnke, R. Klein, and M. Weinmann, "A vr system for immersive teleoperation and live exploration with a mobile robot," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, nov 2019.

[18] C.-Y. Kuo, C.-C. Huang, C.-H. Tsai, Y.-S. Shi, and S. Smith, "Development of an immersive slam-based vr system for teleoperation of a mobile manipulator in an unknown environment," *Computers in Industry*, vol. 132, p. 103502, 2021.

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

[20] S. Livatino, D. C. Guastella, G. Muscato, V. Rinaldi, L. Cantelli, C. D. Melita, A. Caniglia, R. Mazza, and G. Padula, "Intuitive robot teleoperation through multi-sensor informed mixed reality visual aids," *IEEE Access*, vol. 9, pp. 25 795–25 808, 2021.

[21] 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, p. 101170, 2020.

[22] M. Wonsick, T. Keleștemur, S. Alt, and T. Padır, "Telemanipulation via virtual reality interfaces with enhanced environment models," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2999–3004.

[23] 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)*. IEEE, 2018, pp. 1–9.

[24] A. Naceri, D. Mazzanti, J. Bimbo, D. Prattichizzo, D. G. Caldwell, L. S. Mattos, and N. Deshpande, "Towards a virtual reality interface for remote robotic teleoperation," in *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019, pp. 284–289.

[25] 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)*. IEEE, 2018, pp. 5628–5635.

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

[27] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Conference on Robot Learning (CoRL)*, 2018. [Online]. Available: https://arxiv.org/abs/1809.10790

[28] "Pun," https://www.photonengine.com/PUN.

[29] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517.

[30] "Urdf of LoCoBot," https://github.com/facebookresearch/pyrobot/tree/main/robots/LoCoBot.

[31] "ROS#," https://github.com/siemens/ros-sharp.

[32] "RosbridgeServer," https://github.com/RobotWebTools/rosbridge_suite.

[33] A. Aristidou and J. Lasenby, "Fabrik: A fast, iterative solver for the inverse kinematics problem," *Graphical Models*, vol. 73, pp. 243–260, 09 2011.

[34] A. Murali, T. Chen, K. V. Alwala, D. Gandhi, L. Pinto, S. Gupta, and A. Gupta, "Pyrobot: An open-source robotics framework for research and benchmarking," *arXiv preprint arXiv:1906.08236*, 2019.

[35] "Robotic grasping and manipulation competition(iros 2021)," https://rpal.cse.usf.edu/competition$_iros2021/$.