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], Chi-Ruei Wang[1], Po-Lin Li[1], Hsueh-Cheng Wang[1], Haikun Huang[2], and Lap-Fai Yu[2]

***Abstract*— We present Work-from-Home Virtual Reality (WFH-VR), an easy-to-deploy, virtual reality-based teleoperation system for controlling a robot arm. Our system is constructed using a consumer-grade virtual reality device (an Oculus Quest 2) and a low-cost robot arm (a LoCoBot), so it can be easily replicated and set up. Using our system, the user uses a VR controller to control a virtual robot arm to manipulate virtual objects in virtual reality. The joint orientations of the virtual robot arm are sent to the real robot arm located remotely to update its pose for picking up real objects. On the other hand, our system continuously runs the deep object pose estimation on the robot side to estimate the pose parameters of the real objects, which are sent to the virtual reality side to update the poses of the virtual objects. Through these processes, our system achieves a synchronization of the robot arm and objects in virtual reality and those in the real environment.**

We performed experiments to verify that our system can be applied across the globe for VR robot teleoperation with very little latency. We also performed a number of user study experiments to validate the effectiveness of our VR teleoperation system compared to alternative approaches such as keyboard-based teleoperation on a 2D screen and automated deep grasping. The results show that users can perform different object manipulation tasks, such as dining table setting tasks, efficiently and efficaciously using our system.

I. INTRODUCTION

The recent COVID-19 epidemic has affected most countries. It has inspired global efforts to develop robots to mitigate the impacts of the epidemic and to facilitate remote collaboration. In daily scenarios (e.g., eating in a restaurant), the interaction between people is almost inevitable, which increases the difficulty of social distancing and epidemic prevention [1], [2]. In view of such challenges, we investigate how to devise an efficacious and intuitive virtual reality-based teleoperation system for controlling a robot arm to perform tasks, such as for setting a dining table. Akin to some recent efforts on VR teleoperation of robots [3], [4], [5], we are particularly interested in setting up such a system using a consumer-grade virtual reality headset (e.g., Oculus Quest 2) and a low-cost robot arm (e.g., LoCoBot), which are commonly available and can be deployed at scale.

To use our system for teleoperation of a robot arm, the user sits on a chair, wears a VR headset, and holds a VR controller. Through the VR headset, the user sees a virtual environment with virtual objects and a virtual robot arm which resemble the real objects and the robot arm on the real robot side located remotely (e.g., on the other side of the globe). The user uses the VR controller to control the robot arm to perform actions such as moving and grasping. On the other hand, at the robot side, the system constantly runs object pose estimation to estimate the poses of the real objects in the scene, which are transmitted to the VR side for updating the virtual environment to synchronize with the real scene to facilitate visual perception and teleoperation.

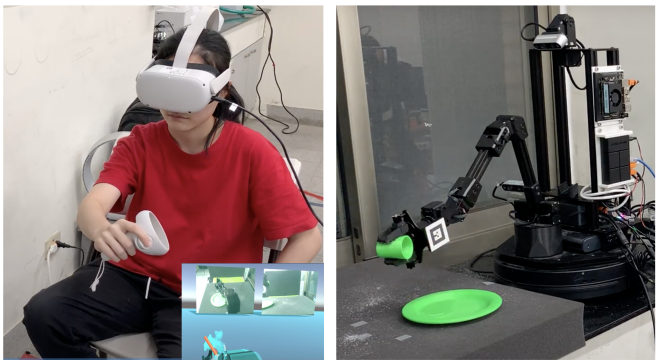

Fig. 1. Our approach enables the remote teleportation of a low-cost robot arm via virtual reality. The user controls a robot arm located remotely to set up a dining table.

In the robotics community, there has been significant attention and competitions [6] on service robots' grasping and manipulation tasks, such as setting up a dining table. In view of this trend, we conducted user evaluation experiments with our system based on dining table setting tasks. Through virtual reality, the users controlled a robot arm to arrange tableware (e.g., plates, knives, forks) to match target positions and to perform tasks (e.g., pouring water). We recorded the success rates and time for performing different tasks, as well as the user's ratings and feedback about using our system.

The major contributions of our paper include the following:

- Proposing a hands-on, easy-to-deploy VR robot teleoperation system based on a consumer-grade virtual reality headset and a low-cost robot arm to facilitate human-robot collaboration. We will release our toolkit to facilitate adoption and extension of the teleoperation system.
- Testing the VR robot teleoperation system across the globe and verifying that the teleoperation can run successfully with little latency.

*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

- Conducting user study experiments to test the VR robot control system for different tasks in a dining table setting and to compare it with alternative approaches such as deep grasping and keyboard control via a 2D screen.

II. RELATED WORK

A. VR Teleoperation

There have been several attempts of VR teleoperation systems to control robot end effectors in 3D space than the ones relying on computer monitors and joysticks/keyboards. Earlier work in [7] used an Oculus Rift to control an industrial robot manipulator that human are not safe around. A ROS RViz plugin for Rift was introduced [8] to enable fully immersive control of a PR2 robot. Subsequently [9] used a VR-teleoperation PR2 to collect human demonstrations for imitation learning. Nevertheless, those systems were not designed for long distance. Recently the ROS Reality [4] was designed to overcome the latency issues over the setup not in a local network by deconstruct/reconstruct point cloud and image data. [3], [5] further carried out motion intent and evaluations and cup-stacking manipulation task using ROS Reality with a hand tracker implementation. Building upon the recent development of state-of-the-art VR system (an Oculus Quest 2) that comes with robust hand controller tracking, our system differs from the above systems in various aspects. We have a long-distance (across the global), but neither point cloud nor raw RGB-D images are sending from robot side. Such design is benefit from the recent success of deep network on object pose estimation [10] and the progress of GPU computing units. We are able to run state-of-the-art algorithms sorely on a low-cost mobile manipulation platform (LoCoBot), and only the estimated object poses computed locally on robot side are sent to remote VR system to a human operator.

B. Low Cost Robotic Platforms

[add papers here]
[11]
[12]
Pinto and Gupta [13] adopted a multi-stage learning approach that combined convolutional neural network and reinforcement learning to learn the grasping pose.

C. Object Manipulation and Regrasp Operations

Robotic pick-and-place is a common task for robotic object manipulation. Based on the perception of the pose of a target object, a robot arm is instructed to perform the corresponding picking actions.

Object affordance is an important consideration for pick-and-place systems. Object affordance reasoning algorithms is highly related to end effector design for object manipulation. To handle clutter, occlusion conditions, and different object geometries, many recent studies have adopted different affordance predictions together with a customized end effector. Hernandez et al. [14] relies on the classic model-based pose-estimation with object registration and the corresponding affordance modes. Zeng et al. [15] have defined four primitives for grasping and suction, and they trained two fully convolutional network (FCN) models to predict the dense pixel-wise affordance probability. Moreover, the affordance, or more precisely, the grasp prediction helps two-finger gripper to execute picking tasks in cluttered environments.

Redmon and Angelova [16] encoded a raw RGB-D image input into several grid cells, and they predicted direct regression to grasp coordinates under the assumption that there was only a single correct grasp per image. Their revised model further predicted multiple grasps per object in real time by using locally constrained predictions. On the other hand, relying on the geometry of an object without color information, Mahler et al. [17] took grasp candidates which were aligned to the depth image as inputs and predicted the probability of grasp success.

Subsequently, two fully convolutional neural networks (FCNs) were used that mapped camera inputs to actions: one FCN pushed with an end effector orientation and location, and the other was used for grasping. Zeng et al. [18] used reinforcement learning to decide whether to push or separate adjacent objects or to pick up objects. All these showed significant progress in solving the picking problem in cluttered environments, but they did not consider placements using desired poses. In our work, instead of relying on trained approaches such as neural networks for picking up and manipulating objects, we devise a VR-based teleoperation system for controlling a robot arm by a human operator. Such a system is particularly useful for performing complex and uncertain tasks which constantly require human's judgement in the object manipulation process.

Recently, researchers have started to apply deep neural networks for 3D object detection and pose estimation. For example, Tremblay et al. [10] introduced the deep object pose estimation (DOPE) method for robotic grasping of household objects. Due to the difficulty of collecting annotated 3D real-world data for training, synthetic data, which is automatically generated, has been used as an alternative for training the neural networks. To ensure that the synthetic data is realistic enough for training, techniques such as domain randomization have been proposed to randomize the training data to introduce variations. In our system, we applied DOPE for estimating object poses at the robot side. Rather than for robotic grasping, the estimated object poses at the robot side are used to update the corresponding virtual objects' poses at the VR side located remotely so as to synchronize the real working environment (at the robot side) with the virtual working environment (at the VR side) to facilitate remote teleoperation.

Due to the limitation of arm kinematic and environmental factors, the end effector of a robot arm cannot be computed using inverse kinematics as the robot arm may not be able execute the computed results. The pose of the target object may need to be changed so that the robot can grasp the object. Regrasp operations proposed by Rohrdanz and Wahl [19] refer to a sequence of pick-and-place operations involved in moving the object from the initial state to the

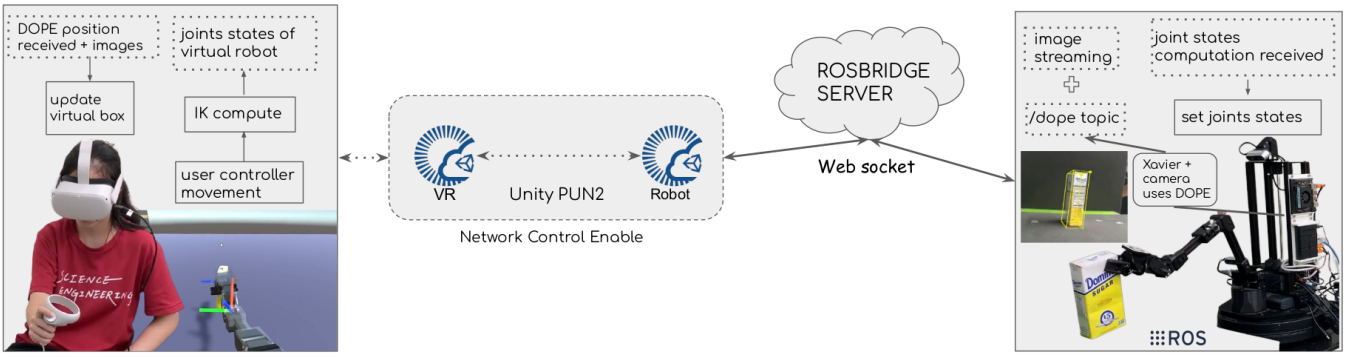

Fig. 2. An overview of our approach.

final state. It is a useful approach to avoid collision with the robot's workspace and to overcome the limitations of kinetics. Xue et al. [20] proposed an automatic planning system for regrasp operations for a multi-fingered hand. The regrasp operations were planned by a breadth-first search using the matrix T from the grasp database in robots during execution. Through control operations and a visual interface synchronized between the virtual and real working environments, our system enables a human operator to control the robot arm smoothly to grasp and manipulate objects.

III. OVERVIEW

Figure 2 shows an overview of our approach. The user and the robot arm are located remotely from each other. To control the robot arm, the user puts on a VR headset for visualization and uses a VR controller for controlling the robot arm. The user sees a virtual environment with virtual objects, which are replicates (by 3D reconstruction or CAD modeling) of the real-world objects at the robot's side. The user also sees a virtual robot arm identical to the real robot arm, having the same joints and dimensions. The user uses the hand controller to control the robot arm and hand to pick up an object and transfer it to another location.

As shown in the figure, the system consists of two major operations flows: to control the robot arm and synchronize the virtual objects in virtual reality with real-world objects. To control the robot arm, the user moves his controller in virtual reality. Based on the end effector's position, our system computes the joint orientations of the virtual robot arm, which, upon confirmation by the user, are sent to the robot side to update the orientations of the real robot arm. To synchronize the virtual objects with the real-world objects, pose estimation is continuously performed on the RGB images captured by the camera attached to the base of the real robot using Deep Object Pose Estimation (DOPE). DOPE estimates the pose parameters (positions and orientations) of the objects in the real scene, which are transferred to the user's side for updating the corresponding virtual objects in virtual reality.

Using this system, the user can see the scene in virtual reality synchronized with the object settings in the real world at the robot side. The user can remotely control the objects in the real scene intuitively and conveniently through virtual reality. Note that all the data transmitted (robot joint orientation parameters, object pose parameters) is minimal in size so that instant control and synchronization between the real and virtual scene is achieved, making the system more intuitive and convenient to use.

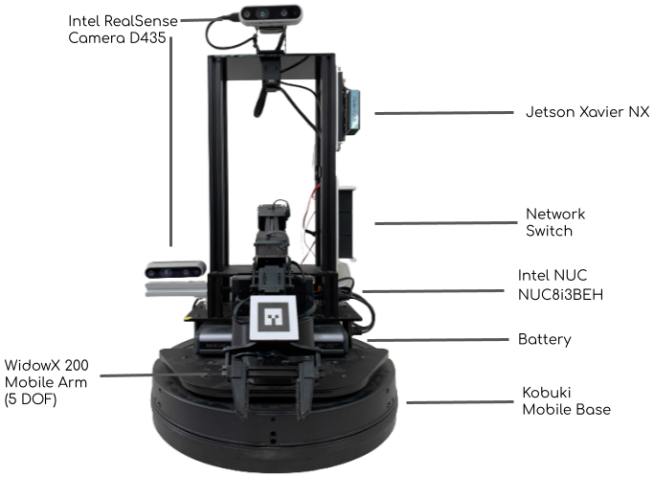

Fig. 3. The robot used in our setup.

IV. TECHNICAL APPROACH

A. Hardware

Our teleoperation platform is based on a consumer-grade VR device, the Oculus Quest 2, which costed about $420, and a low-cost robot arm, LoCoBot, which costed about $5,000. For VR devices, the Oculus Quest 2 system provides a headset for head-mounted display, which is a singular fast-switch LCD panel with a $1,832 \times 1,920$ per eye resolution and runs at a refresh rate of up to 120 Hz. It also comes with two Oculus Touch hand controllers, each with 6 DoF pose tracking with infrared LEDs, which allows the controllers to be fully tracked in 3D space by the Oculus Quest 2 constellation system.

Figure 3 shows a picture of our robot. We used the LoCoBot, a low-cost mobile manipulator robot, for our purpose This robot consists of 5 main components crucial for the control system: Intel RealSense RGB-D Camera D435, a

Jetson Xavier NX, WidowX 200 Mobile Arm (5 DOF), Intel NUC, Kobuki Base. In addition to the original camera at the top of the robot, we also attached another Intel RealSense RGB-D Camera D435 at the near arm base position. The camera at the base is used to capture RGB images for pose estimation of objects by DOPE. Note that we do not use the original camera at the top of the robot to capture images for running DOPE due to the high possibility of robot arm occlusions, which can cause pose estimation failures. Our teleoperation system is written in Unity, a 3D game engine that supports major VR headsets such as the Oculus.

We propose a three-part solution to design our system to control the robot in Virtual Reality. First, we use a Deep Object Pose Estimation (DOPE) [10] system to obtain the poses of the real objects for updating the poses of their virtual object counterparts in the virtual reality view. Second, our system at the VR side computes the inverse kinematics solution for updating the robot arm's joint orientations, shows a visual interface for the user to peform teleoperation, and sends the joint orientations to the real robot at the remote robot side. Lastly, our network control component uses the Photon Unity Network 2 (PUN 2) framework to transmit the robot joint orientation parameters to control the real robot arm, and the estimated object pose parameters to synchronize the virtual and real working environments.

B. Pose Estimation

We briefly describe the DOPE [10] approach to estimate an object's poses, which is combined with the robot's position and orientation to produce a corresponding image of the rendered view in virtual reality. First, DOPE uses a deep neural network that estimates belief maps of 2D key points of all the objects in the image coordinate system from the LoCoBot's RealSense D435 camera attached to the robot's base. Note that only the RGB image, not the depth information, is used for pose estimation. Second, peaks from these belief maps are fed to a standard perspective-n-point (PnP) algorithm [21] to estimate the 6-DoF pose of each object instance.

The physical LoCoBot is equipped with two cameras, one at the top of the base, another at the bottom right of the base as shown in Figure 3 which we have the exact position and orientation at any given moment. We utilize the top camera for real-time image streaming from robot to Virtual Reality. Side camera are response for capturing images for pose estimation of the objects. The robot is also equipped with a NUC computer responsible for the communication between the robot and Unity and Xavier NX for running DOPE calculation. The NUC is connected to Rosbridge Master IP, an open-source program that converts JSON API to ROS functionality, which has the cameras' coordination topic and DOPE's object topic readily to listen by Unity.

In the Unity program, we use the ROS# package, open-source software libraries and tools in C# for communicating with ROS from .NET applications, connect with ROS Master IP and listen to the related topic of the robot's orientation and DOPE object's orientation in Unity. For object rendering (e.g., a Domino sugar box), we leverage the popular YCB object model for user reference.

C. Robot Arm Control in Virtual Reality

1) Visual Interface: In order to effectively control the robot in the virtual reality interface, the user must have a good reference about the surrounding in the real working environment

First, as mentioned above, after the system at the robot side has received the position and orientation data from DOPE and the robot itself, it set the virtual objects and the virtual robot arm at the VR side accordingly to synchronize with the real-world settings. However, self-occlusion sometimes occurs when the robot arm is picking up objects. In that case, the DOPE method may fail to estimate the objects' poses. To mitigate this problem, our visual interface at the VR side uses a rigid body to represent the picked-up virtual object when DOPE fails to estimate the pose of the real object at the robot side (e.g., due to self-occlusion). The rigid body is attached to the robot arm so that its virtual object's pose is updated following the movement of the virtual robot arm controlled by the user. In other words, the visualization provided by the rigid body acts as a backup plan in case of self-occlusion. In this case, our system renders the object as mild transparent, indicating that the pose of the object is not certain.

Second, to further guide the user in virtual reality, we take advantage of the robot's top camera to stream the RGB video of the robot side (i.e. the tabletop) to a screen in the virtual working environment. This will enable the user to have a better understanding and reference of where the objects are and what is happening at the real robot side.

Third, the user should know the robot arm's current position and orientation, and where it will move to. We take advantage of the open-source LoCoBot's URDF model by Trossen Robotics to recreate the virtual robot model inside Unity. According to the joint state's data sent from the robot to the ROS Master IP, the virtual robot model's joints and position will be updated several times per frame, letting the user know the current status of the real robot arm.

2) Control Interface: To minimize latency in communication and avoid expensive computation at the robot side that uses a NUC, we use an inverse kinematics solver (based on the FinalIK package) in Unity to compute the physical robot's joints' positions and orientations on the end effector of the virtual LoCoBot. The Oculus Quest's right controller's position and orientation are mapped as the end effector target of the virtual robot, allowing the user to move the arm freely to guide the movement of the virtual robot arm, which is visualized in real-time. After the user confirms the desired position of the virtual robot's arm, by pressing the grip button on the controller, the computed robot arm joints orientations and positions topic will be sent to the ROS Master IP, which then triggers a handler script at the robot's NUC to set the joints of the real robot arm accordingly and accurately.

D. Network

Photon Unity Network 2 is a real-time cloud framework that host multiplayer games for developer on Unity. By

leverage the usage, we can send the position and orientation data of the virtual robot over the network which can be received from the physical robot's host. Since only the physical robot side need to be connected to Master IP, the VR user can be virtually anywhere in the world can able to control the robot.

V. EXPERIMENTS

Our primary goal is to investigate the effectiveness of the VR robot teleoperation system. We want to investigate the user experience in teleoperating a robot arm (i.e. a LoCoBot) across the globe and also in performing some common dining table tasks.

A. VR Robot Teleoperation across the Globe

We tested the proposed VR teleoperation platform between two universities across the globe. The VR side was set up at George Mason University in Virginia, USA. The real robot side was set up at National Yang Ming Chiao Tung University in Taiwan.

The user in USA put on a VR headset and controlled a LoCoBot robot arm in Taiwan. The user was instructed to pick up a box and place it at a target position. The user saw a virtual box and a virtual robot arm which resembled the real box and the real robot arm in Taiwan. In addition, a live video showing what was happening at the robot side was streamed to a virtual screen in the virtual working envionment to facilitate teleoperation.

The user reported no awareness of the latency of the robot arm control. The user commented that the pose update of the virtual box (synchronized with the real box) helped him a lot in manipulating the object as he could intuitively judge the current status of the box as he teleoperated the robot. There was about one second of latency in streaming the video, yet the latency did not adversely affect the user's judgement and performance as the manipulation motion was not done at a very fast pace. The results showed that our system could support across the globe manipulation with little latency and the DOPE pose estimation facilitated teleoperation.

B. Dining Table Experiments

The experiment tasks are inspired by the IROS 2021 Robotic Grasping and Manipulation Competition Service Robot Track. We chose a range of tasks and modified them to fit with our robot setup. Figure 4 depicts the five tasks. The objects we used in the experiments are inspired by the YCB objects and common objects on a dining table. Due to hardware limitations, we chose to scale down the objects by modeling CAD models followed by 3D printing. We describe the tasks as follows:

- Task 1: pick and place plates and a bowl. This task required the robot to pick up plates and a bowl from an initial position and then stack them up in order. This is challenging because it is easy to knock down an object placed previously. We regard it as a success if the plates and bowl are stacked in order as shown in the target state.
- Task 2: pick and place tablewares. This task required the robot to pick up the tablewares and rearrange them to match the target state. This is challenging because the tablewares are flat. The user also needed to put them to surround a small plate closely. We regard it as a success if the tablewares are arranged in the right relative positions as in the target state.
- Task 3: pick and place glasses and cups. This task required the robot to pick up a tea cup, a wine glass, a mug, and a small plate from their initial positions and rearrange them to match the target state. It is challenging as the robot could easily knock down an object placed previously. The tea cup also needed to be put on a small plate. We regard it as a success if all objects are arranged in the right relative positions.
- Task 4: transport sugar grains. This task aimmed to grasp a small spoon to scoop sugar grains from the bowl and then place the grains on a big plate. The scooping action is a natural human action. This task is challenging due to the robot hardware limitations in the degree of freedom [what DOF limitation is there?]. A successful trial means that the sugar grains on the plate weighs more than 1g.
- Task 5: transport liquid. This task aims to grasp a small water cup to pour water-like liquid [is this water?] onto the big plates. Pouring action is a natural human action. This task is challenging due to hardware limitations in the degree of freedom. A successful trial means that the water poured onto the big plate weighs more than 5g.

Experiment Setup. We used the deep grasping model provided by PyRobot [11] to detect the grasping point of the object. Then we placed the object at the position to be grasped for the first grasping and recorded the trajectory of the arm movement. Then we placed the object where it should be placed, and record the trajectory that must be moved for the second grasping. Then we connected the record of the first paragraph and the second track after reversal to play together to achieve the task of making pick&place tableware. [not sure what you mean, need rewrite]

We invited 12 users to use our VR-LoCoBot-teleoperation system to do the tasks. This was the first time the users used such a system. The users could control the robot arm to interact with the objects using different strategies based on different primitive motions such as pushing, grasping, placing, and any action that the user would like to use. Each task was given a 5 minute limit to perform. There was no limit in the number of interactions between the robot arm and the object.

We also invited a researcher experienced with our VR-LoCoBot-teleoperation and remote keyboard teleoperation provided by PyRobot to do the tasks.

Result and Discussion. We describe the results of the experiments.

1) *Deep Grasping Method.* But in this part, we can only do pick&place in task2 with deep grasping. We originally expected task1 3 to be able to use deep grasping for gripping, but in fact, it failed after the

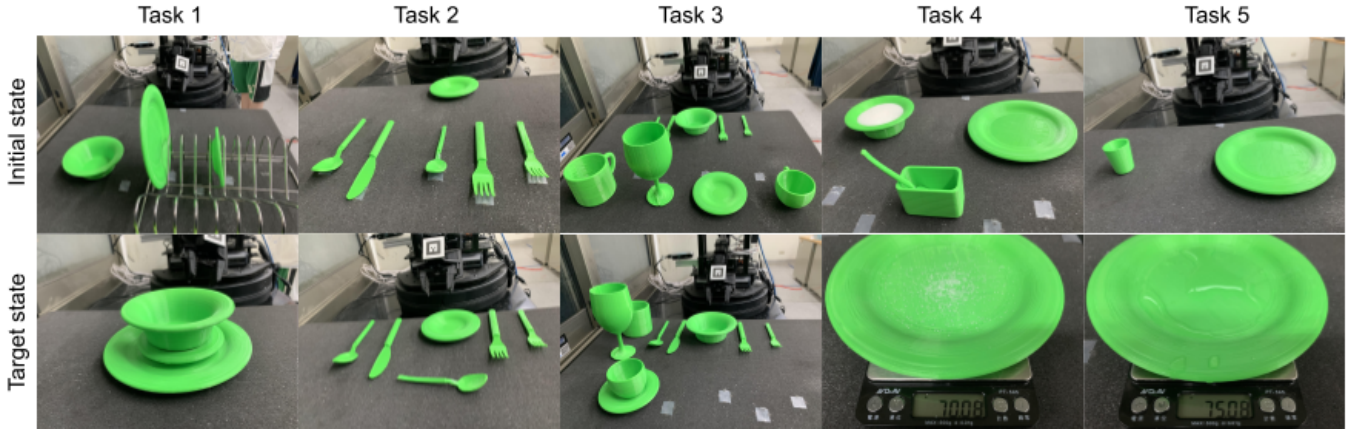

Fig. 4. The five tasks of the dining table experiment.

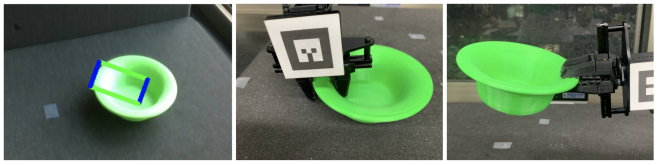

Fig. 5. Left: gripping point predicted by deep grasping. Middle: gripping performed by the LoCoBot. It failed to grip the rim of the bowl due to its special shape. Right: gripping the bowl by VR teleoperation.

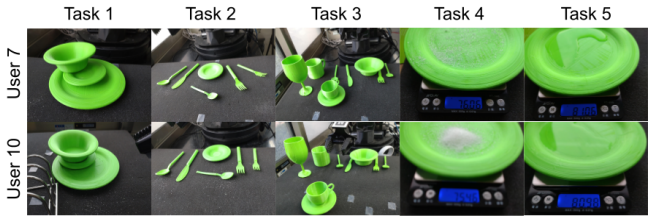

Fig. 6. User performance in dining table for each task.

test. The reason is that the data used by deep grasping was not used as a label for tableware. For LoCoBot, to be able to clamp the tableware, only a very specific clamping angle can be used, which is very different from the original deep grasping strategy as long as the object can be clamped.

2) *The 12 Users using VR-LoCoBot-Teleoperation.* Figure 6 shows the target states reached by some users. They were very surprised that they could use the VR control arm to complete the task. The fluency and delay of the VR control robot exceeded their expectations. The completion of the task through VR is easier than they thought. Compared with deep grasping, the user's operation is more flexible and more flexible. Many, for example, when the knife and fork are not aligned, people can easily correct small angles by moving and moving. The moving process is not as old-fashioned as an automated arm. You may use drag to move the object It is a labor-saving way to move to the correct position, and even so, there will still be failures. The main reason is that the user is operating for the first time and is not familiar with the system, and the degree of freedom of the arm is only five axes, unlike Human arms are so handy to operate, but the subjects also mentioned that even so, they still have a pretty good experience, and they look forward to more progress in the future and they are willing to experience it again. As shown in Table I,
3) *The researcher using VR-LoCoBot-Teleoperation or Keyboard.* As shown in Table II,

TABLE I

SUCCESS RATES AND AVERAGE TIME OF OUR VR-LOCOBOT-TELEOPERATION OVER THE DINING TASKS BY 12 USER.

	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**
Success rates	11/12	12/12	10/12	11/12	12/12
Average time (mins)	1"30				

C. Comparison of the Results of Different Approaches

In this section, we compare...

VI. LIMITATIONS AND FUTURE WORK

There are several major limitations of our system, which inspire interesting directions for future work.

First, we used a camera attached to the base of the robot for capturing the current scene for pose estimation of the available objects. When the robot arm picks up an object, the pose estimation of the object by DOPE may fail due to self-occlusion by the robot arm. Future work may consider attaching additional cameras to the robot or the environment to capture multiple images for pose estimation to make the system more robust to self-occlusion.

Second, our current system supports only one robot arm. It would be interesting to extend it to support two robot arms so that the operator can use both of his hands to pick up or manipulate objects as in the real world. Such a dual arm model could potentially enable more sophisticated object manipulation, yet would bring up new technical challenges such as self collision and higher chances of self-occlusion that would need to be addressed.

TABLE II

SUCCESS RATES AND AVERAGE TIME OF DIFFERENT TELEOPERATION OVER 5 TRIALS OF THE DINING TASKS BY 1 USER.

	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**
Keyboard	4/5 (1"00)	5/5 (1"00)	5/5 (1"00)	5/5 (1"00)	5/5 (1"00)
VR					

TABLE III

COMPARISON OF THE RESULTS OF DIFFERENT APPROACHES ON TASK 2.

Method	**Trial**	**Success**	**Time**
Deep-grasping	10	60	60
Keyboard Teleoperation	10	60	60
VR- Teleoperation (Only 2D plane)	10	60	60
VR- Teleoperation (DOPE)	10	60	60
VR- Teleoperation (MaskRCNN+ICP)	10	60	60

Third, we used a VR controller for controlling the robot hand which has only two fingers acting as a gripper. Replacing the robot hand with a five-finger model would allow more complex pickup and object manipulation motions. In this case, it seems reasonable to replace the VR controller with a glove that supports hand-tracking. The user will then be able to use his hand directly to control the robot's hand at the remote site. Paired with force feedback by the glove, it may lead to more intuitive, natural, and accurate hand control by the user.

Finally, we only experimented with rigid, non-deformable objects using our system. As many objects in the real world such as paper, clothes, and plastic bags are deformable, it would be worthwhile to extend our system to deal with deformable objects. One approach is to keep track the object shapes continuously which will be fed to the virtual environment to update the shapes of the virtual objects accordingly. Another approach is to capture the 3D geometry of the objects in real time which is sent to the operator side for updating the visualization, yet this approach will be demanding in terms of the bandwidth. We are interested in investigating effective system designs to handle such problems.

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] E. Rosen, D. Whitney, E. Phillips, G. Chien, J. Tompkin, G. Konidaris, and S. Tellex, "Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays," *The International Journal of Robotics Research*, vol. 38, no. 12-13, pp. 1513–1526, 2019.

[4] D. Whitney, E. Rosen, D. Ullman, E. Phillips, and S. Tellex, "Ros reality: A virtual reality framework using consumer-grade hardware for ros-enabled robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[5] 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.

[6] 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.

[7] IVRE – an immersive virtual robotics environment. [Online]. Available: https://cirl.lcsr.jhu.edu/research/human-machine-collaborative-systems/ivre/

[8] PR2 Surrogate. [Online]. Available: http://wiki.ros.org/pr2_surrogate

[9] 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.

[10] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790*, 2018.

[11] 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," *CoRR*, vol. abs/1906.08236, 2019. [Online]. Available: http://arxiv.org/abs/1906.08236

[12] A. Gupta, A. Murali, D. Gandhi, and L. Pinto, "Robot learning in homes: Improving generalization and reducing dataset bias," *arXiv preprint arXiv:1807.07049*, 2018.

[13] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3406–3413.

[14] C. Hernandez, M. Bharatheesha, W. Ko, H. Gaiser, J. Tan, K. van Deurzen, M. de Vries, B. Van Mil, J. van Egmond, R. Burger *et al.*, "Team delft's robot winner of the amazon picking challenge 2016," in *Robot World Cup*. Springer, 2016, pp. 613–624.

[15] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Dafle, R. Holladay, I. Morona, P. Q. Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2018.

[16] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1316–1322.

[17] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust robot vacuum suction grasp targets in point clouds using a new analytic model and deep learning," *arXiv preprint arXiv:1709.06670*, 2017.

[18] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," *arXiv preprint arXiv:1803.09956*, 2018.

[19] F. Rohrdanz and F. M. Wahl, "Generating and evaluating regrasp operations," in *Proceedings of international conference on robotics and automation*, vol. 3. IEEE, 1997, pp. 2013–2018.

[20] Z. Xue, J. M. Zoellner, and R. Dillmann, "Planning regrasp operations for a multifingered robotic hand," in *2008 IEEE International Conference on Automation Science and Engineering*. IEEE, 2008, pp. 778–783.

[21] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.