Multi-view Merging for Robot Teleoperation with Virtual Reality

Dong Wei[1], Bidan Huang[2†], Qiang Li[3]

***Abstract*—In robotic teleoperation, an operator usually needs to be trained for long hours. One of the factors leading to this steep learning curve is the quality of telepresence. This paper proposes a novel and user-friendly telepresence interface for manipulation. Visual information from different views is merged and presented to the operator in an intuitive way to facilitate the task based on state-of-the-art virtual reality technology. Besides rendering the scene, a virtual robot is also rendered in the immersive view so that the robot is visible even when it is occluded. We performed a series of user studies to evaluate this interface. The results show that the proposed interface can achieve a better task performance compared to a standard approach in both the manipulation efficiency and the users' preferences.**

***Index Terms*—Telerobotics and Teleoperation; Virtual Reality and Interfaces**

I. Introduction

Building a fully autonomous robot has long been one of the goals of robotics research. However, even state-of-the-art autonomous robotic systems fall short in many aspects. Teleoperation remains a more practical approach, as it empowers remote robots with the knowledge and skills of human operators. A typical teleoperation system is comprised of a leader device, controlled by the human operator, and a follower device, which is the remote robot (Fig. 1(a) and (b))[i]. Information captured by cameras and sensors on the follower is relayed back to the leader device, providing the operator with the necessary feedback to complete tasks such as search and rescue [1], tunnel inspection and repair [2], as well as robotic-assisted surgery [3].

A highly immersive and intuitive user interface, capable of providing multi-modal sensory feedback to the operator with minimal latency is essential for effective robotic teleoperation. Among other stimuli such as haptics and audio, visual feedback is arguably the most important sensory feedback for telepresence, as it enables the operator to localize the robot and understand the remote environment [4]–[6]. Efforts to improve the quality of visual feedback has been an important theme in research. One of the most challenging issues with visual feedback is the presence of occlusion in a cluttered environment. This is especially pertinent when cameras on the remote robot are mounted at a single site, thus restricting the number of perspectives they can capture. Alternatively, cameras can be placed on multiple locations of the robot. A typical multi-camera setup involves global and local cameras, where a static global camera provides a macroscopic view of the workspace [7]–[9], while a local camera mounted on the robot end-effector captures a more detailed local view [5], [10]. In such a configuration, the way in which multiple perspectives are presented to the operator will play a critical role in the user experience.

A common approach for multi-view merging is Picture-In-Picture (PIP), where video streams from multiple views

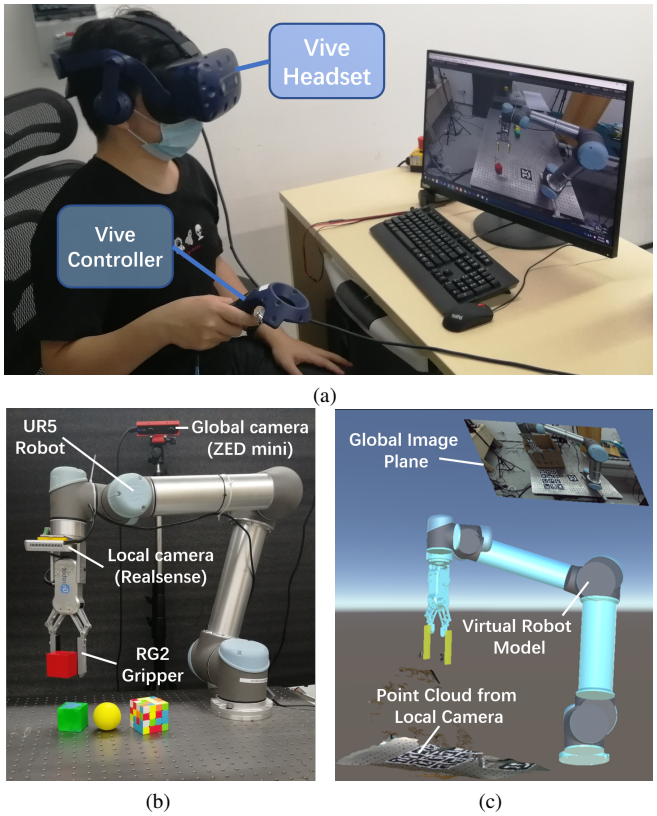


Fig. 1: System Design (a) Leader device includes Vive headset and controller; (b) Robotic follower includes UR5 robot, RG2 gripper, global and local camera; (c) the VR system in Unity includes global image plane, virtual robot model and point cloud from local camera.

Manuscript received: April, 28th, 2021; Revised July, 23rd, 2021; Accepted August, 23rd, 2021.

This paper was recommended for publication by Editor Jee-Hwan Ryu upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Tencent Robotics X

† denotes the corresponding author.

[1] D. Wei is with State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou, China (email: rexyuroyuro@gmail.com)

[2] B. Huang is with Tencent Robotics X, China (email: bidanhuang@tencent.com)

[3] Q. Li is with Neuroinformatics Group, Cluster of Excellence Cognitive Interaction Technology (CITEC), Bielefeld University,33619 Bielefeld, Germany (email: qli@techfak.uni-bielefeld.de.)

Digital Object Identifier (DOI): see top of this page.

[i]"leader and follower" is used here instead of the conventional "master and slave" to avoid the concern of association to racism and human subjugation.

are displayed concurrently and placed next to each other [4], [11], [12]. While PIP is straightforward to implement, it has two disadvantages. Firstly, it requires the operator to switch between multiple 2D views to acquire and understand the whole 3D scene of the remote site. This could significantly increase the cognitive burden. Secondly, the local view is attached to the robot end-effector frame, which moves with the robot. Operating under a moving frame is not natural for human.

Another popular technique is to reconstruct the entire 3D scene from multiple point clouds. This approach necessitates the use of multiple depth cameras and complex algorithms to fuse the information together [13], [14]. Rendered to the user through a Virtual Reality (VR) interface, the rich sense of depth and color has been shown to improve operation efficiency in various studies [4], [15]. The downside of this approach is the huge bandwidth required to transmit multiple point clouds, as well as the high computational cost to merge them, resulting in poor real-time performance that impairs the user experience.

There are numerous works in the literature that try to achieve good visual experience while keeping computational requirements manageable. Theofilis et al. [16] proposed a panoramic reconstruction approach in the VR interface that incorporates robot head control. Kohn et al. [14] merged point clouds from only two global external cameras, thus reducing visual occlusion while remaining computationally tractable. Omarali et al. [17] proposed to use OctoMap to generate a lightweight map of the robot's workspace from solely in-hand camera data, arguing that global information of the environment is not essential for task completion. On the other hand, Chen et al. [18] adopted a brute force approach, leveraging large servers to achieve real-time 3D reconstruction from multiple point clouds. It is apparent that despite the many solutions explored, an intuitive way of rendering multiple perspectives in real-time with minimal computational power remains elusive.

In this work, we present a novel telepresence approach to merge visual information from multiple cameras based on a VR interface. We utilize a static global stereo camera and a local RGB-D camera mounted on the end-effector of the robot. The global view is displayed to the operator as a stereoscope image, thus negating the need for computational 3D reconstruction. The local view is aligned and superimposed into the global view as a 3D point cloud, allowing the operator to view and operate from the same perspective. Our preliminary work showed that our approach can provide the operator with essential 3D information for manipulation tasks across multiple scales [19] while remaining computationally lightweight.

In this paper, we improved the quality of the multi-view merging through the use of a dynamic online re-calibration algorithm. We also conducted a comprehensive user study to examine the performance of the proposed system qualitatively. Different telepresence modes were evaluated with respect to their pros and cons with objective and subjective analysis. Experimental results showed that the proposed interface outperformed the existing approaches in efficiency and usability.

The main contributions of this paper are listed as follows:

1) We proposed a lightweight multi-view merging approach to render multiple viewpoints. Using this approach, the operator can manipulate with a natural viewpoint and can "see through" the occlusions. When necessary, the operator is able to change its view angle to observe the objects from different perspectives.
2) We implemented an online, dynamic vision-based re-calibration technique that aligns the global stereo images, the local point cloud and the robot virtual model in an immersive environment.
3) We conducted an in-depth efficiency analysis of the user study in complicated manipulation task and compared our proposed working modes with the PIP mode in a teleoperation system.

The rest of the paper is structured as follows. Session II describes the proposed system. Session III explains the design of the experiments and Session IV details the user study. Conclusion and future work are described in Session V.

II. Augmented multi-view teleoperation interface

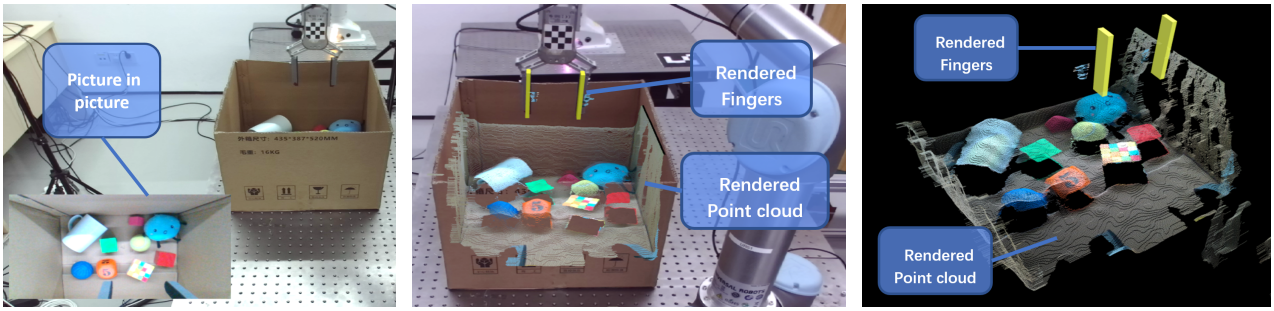
In this section, we detail the design of the proposed system which is a multi-view teleoperation interface for manipulation tasks. As shown in Fig. 1, the follower is a robotic arm with a fixed global camera and a local camera mounted on the end-effector, and the leader is a VR system. The human operator wears the VR headset and controls the robot via the VR controller (Section II-A).

In this multi-view system, the global camera provides an overview of the surrounding, and the local camera provides a close perspective of the manipulation scene. To alleviate the aforementioned problems of the PIP approach (Fig. 2(a)), we have developed a novel approach for merging the global and local views: the Point Cloud Projection (PCP). In PCP, we render the global stereo view and project the local 3D point cloud to it, allowing the operator to see them from the same static perspective. In this way, the operator can be fully aware of both the global and local situations. Manipulation can be performed intuitively as all the views are aligned to the operator's frame, i.e. the global camera's frame. As a result, this creates a "see-through" feeling when the manipulation scene is occluded by the environment, allowing the operator to manipulate easily in this difficult situation (Fig. 2(b)).

Rendering the global view with the stereoscopic approach requires a smaller memory and lower computational cost[ii] comparing to the point cloud based approaches [18]. We only render the local scene with point cloud to provide the essential information for manipulation. To work with the PCP, we have also developed a Point Cloud Inspection mode (PCI). In PCI, the operator is able to control the view angle and zoom in and out of the point cloud through natural head movement (Fig.2(c)). A virtual robot model is also rendered in the view for the occlusion problem.

To summarize, the visualization of the scene contains four main components which all need to be integrated into the system:

[ii]In our work, a pair of stereo image is around 2.7MB, while the corresponding point cloud sizes about 28MB.



(a) (b) (c)

Fig. 2: Three different teleoperation interfaces. (a) Picture in Picture (PIP). (b) Point Cloud Projection (PCP). (c) Point Cloud Inspection (PCI).

- VR stereo rendering of the global view (Section II-B)
- Virtual robot model (Section II-C)
- PIP rendering of the global and local views (Section II-D1)
- 3D PCP rendering of the global and local views (Section II-D2)
- 3D PCI rendering of the local view (Section II-D3)

A. System Setup

In this proposed system, the follower (Fig. 1(a)) consists of a 6 d.o.f robot arm (UR5[iii]) mounted with a two finger robot gripper (RG2[iv]), a global view stereo camera (ZED mini[v]) installed on a fixed location related to the robot base and a local view depth camera (Realsense D415[vi]) attached to the wrist of the robot. The relative poses between the cameras and the robot are firstly estimated via an offline hand-eye calibration process and fine-tuned online via a dynamic calibration process detailed in the Section II-D2. We elongate the robot fingers such that the finger tips can be seen by the local depth camera, and has a minimum working distance of $11cm$.

For the leader (Fig. 1(b)), we use a commercially available VR system HTC Vive [vii]. This system consists of a head mounted display (HMD) and two controllers. The HMD has two displays, one for each eye. Visual 3D effect can be created by presenting two images with parallax to each display. The Vive system tracks the human head movement via sensors on the HMD and the hand movement via the controllers. The human operator uses the controllers to control the movement of robot. The open/close status of robot gripper is coded in binary and controlled by a button on the controller. To enhance the user experience, the relative pose between the global camera and the robot arm mimics the pose between the human head and the right arm.

We use the Unity game engine for scene rendering and present the robotic operation scene to the user via the VR interface. With this engine we are able to provide a high quality immersive experience to the user through a HTC Vive. In the VR environment (Fig. 1(c)), a virtual follower system is firstly built and the main components include the virtual global and local cameras and a virtual robot model. To resemble the real system, the virtual global camera and the virtual robot have fixed locations, and the virtual local camera is fixed to the frame of the robot wrist. During the teleoperation, the virtual robot mirrors the real robot motion.

[iii] https://www.universal-robots.com/
[iv] https://onrobot.com/en/products/rg2-gripper
[v] https://www.stereolabs.com/zed-mini/
[vi] https://www.intelrealsense.com/depth-camera-d415/
[vii] https://www.vive.com

B. VR Rendering of Global View

The global camera ZED mini is a stereo camera, of which the distance between the left and the right camera follows closely to the pupillary distance of human. Wearing the HMD, the operator can have a natural 3D sense of the robot's surrounding environment. To this end, the left and right global images are rendered onto individual frames in front of the virtual global camera, which are placed at a plane to match the file-of-view (FOV) of the camera inputs. Each eye is only able to see the plane that represents the view from the camera for that eye. An intuitive 3D sense of the operation scene is hence constructed to the user via the disparity of the two displayed images.

C. Virtual Robot Model

During teleoperation, the robot arm can be occluded and this increases the difficulty of operation, especially when the gripper is out of view. To tackle this, we place a virtual robot model in the VR to show the occluded parts of the robot. The URDF model of the robot is imported to Unity with the pose aligned to the real one.

To render the scene with the correct depth and occlusion effect, we exploit the depth information. Both the global and local cameras can provide the depth value on their pixels. In VR, the physical depth is converted to the screen space depth of each texels, i.e. the texture pixel. Screen space depth is a value ranging from zero to one, with a value of zero representing the smallest distance in front of the camera that can be rendered and the value of one representing the maximum distance that will be rendered. This allows the engine to compute the occlusion and decide what to be displayed to the user. By having both the scene and the geometric objects in the same space we can determine which object is occluded in reality and present the corresponding effect to the user in the virtual scene. This enables the user to understand the motion

of the robot as it moves through complex and occluded scene or handle self-occlusion by the robot during difficult tasks.

D. Local View Visualization

In order to provide more information about the operation scene, a local camera mounted on the robot wrist is used to capture the scene from a different view. This view is essential when the operation scene is occluded from the global view. Since the local camera is close to the gripper, more details about the robot and the manipulated objects can be observed. As the robot moves, the local camera moves accordingly to the gripper and hence the gripper is always in the FOV. The local camera provides a single RGB image along with the depth from its infrared cameras. We implemented both the PIP and the PCP to visualize the local view and conducted user studies to evaluate their performance. In the following paragraphs, we detail these two settings.

1) Picture in Picture Mode: PIP is the standard practice to display the local view. In this mode, the user can see the local view from a 2D image overlayed at the corner of the global view (Fig. 2(a)). This picture can be toggled on and off. For manipulation, a part of the fingertip can be seen and it allows the user to decide the robot motion. However, when the local camera moves with the robot, the global-local camera relative pose changes too. The user hence has to constantly switch between the global and local views to figure out the global-local relation. Note that the robot movement direction does not map the local view moving direction, which can increase the difficulty of teleoperation (more details in the Section III).

2) Point Cloud Projection Mode: In this mode, the RGB-D local image is firstly converted to a colored point cloud. The point cloud is then projected to the global image planes and allows the user to see the local scene from the global view. In this approach, the user does not need to switch between two views but still has a comprehensive understanding of the whole operation scene. When the local scene is occluded from the global view, e.g. when picking up an object from a large box or plugging in a cable to the back of a computer, the user can effectively see through the obstructions and operate the robot naturally. This mode maximizes the user's ability to understand the scene in terms of depth, color and relative position.

In the PCP mode, it is essential to align the local view with the global view so that the objects observed from the local camera can be correctly displayed and hence it allows the user to make correct decision. This is to say, the pose between the local camera and the global camera needs to be computed precisely in real time during the entire teleoperation. To this end, we firstly conducted offline hand-eye calibration[viii] to calculate the relative pose between the global camera and the base of the robot (${}^{r}H_{g}$), the local camera and the robot end-effector (${}^{ee}H_{l}$). With the robot kinematics we can access the end-effector pose in the base frame (${}^{r}H_{ee}$), and the global-local camera relative pose (${}^{g}H_{l}$) can be computed as:

$$ {}^{g}H_{l} = ({}^{r}H_{g})^{-1} \cdot {}^{r}H_{ee} \cdot {}^{ee}H_{l} \tag{1} $$

Algorithm 1 Dynamic Calibration

Initialize ${}^{g}H_{l}$ with Equation 1
repeat
  Read Global left RGB image, Local RGB-D image
  Detect 2D features F_{global}, F_{local}
  Compute 3D locations P_{local} with F_{local}
  Transfer P_{local} to global camera base P^{*}_{global}
  Project P^{*}_{global} to 2D feature points F^{*}_{global}
  if $dist\{F_{global}, F^{*}_{global}\} \leq thresh$ **then**
    keep F_{global},F^{*}_{global},P_{local}
  else
    remove F_{global},F^{*}_{global},P_{local}
  end if
  ${}^{g}H'_{l} \leftarrow$ SolvePnP$\{F_{global}, P_{local}\}$
  $D' \leftarrow \sum dist\{F_{global}, F^{*}_{global}\}$ with ${}^{g}H'_{l}$
  $D \leftarrow \sum dist\{F_{global}, F^{*}_{global}\}$ with ${}^{g}H_{l}$
  if $D' \leq D$ **then**
    ${}^{g}H_{l} \leftarrow {}^{g}H'_{l}$
  end if
until exit

This offline calibration provides us an initial global-local cameras relative pose. However, in practice we found that ${}^{g}H_{l}$ is not always a constant. It is difficult to have a prefect hand-eye calibration, and the local camera pose ${}^{ee}H_{l}$ can change slightly due to the movement of the robot. As a result, the local view may gradually misalign with the global view over time. To tackle this, a dynamic calibration process is run during the teleoperation to adjust the global-local camera relative pose ${}^{g}H_{l}$ online and ensure the two views are always aligned.

The dynamic calibration is formulated as a perspective-n-point (PnP) problem (Algorithm 1). Using the left images on camera and the local camera 2D image features via SIFT are detected and matched. We denote the matched feature pairs as F_{global} and F_{local}. The 3D locations of the F_{local} are computed according to their depth images and resulted in a set of 3D feature points P_{local}. With the F_{global} and P_{local}, the new global-local camera relative pose ${}^{g}H'_{l}$ is estimated by solvePnP[ix]. We hence update the local camera pose ${}^{ee}H_{l}$ by Equ. /citeeqn:pose with the current end-effector pose ${}^{r}H_{ee}$ and the fixed local camera pose ${}^{r}H_{g}$.

In practice, the feature matching can result in mis-matches and reduce the accuracy of ${}^{g}H'_{l}$. To tackle this, we apply extra steps to improve the feature matching. With the latest global-local camera relative pose, 3D points P_{local} are transferred to the frame of the global camera (P^{*}_{global}). Projecting these points to the global left image gives a set of 2D feature points F^{*}_{global}, each of which corresponds to the detected F_{global}. The 2D Euclidean distance between the detected feature locations and their corresponding projected feature locations are hence computed. Feature pairs with large distance are considered as mis-matches and are omitted. The remaining feature pairs are kept as good matches and their distances are summed to a total distance D. The corresponding good F_{global} and P_{local} pairs are used for the solvePnP. With the result ${}^{g}H'_{l}$, a new

[viii] http://wiki.ros.org/rc_visard/Tutorials/HandEyeCalibration

[ix] https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html

total distance D' is computed. Finally, the pose is updated to the new value ${}^{g}H_l'$ when distance D' is smaller than the D. Here, we presume that the difference between ${}^{g}H_l$ and ${}^{g}H_l'$ in two consecutive iterations is small, which holds well in our experiments.

3) Point Cloud Inspection Mode: For fine manipulation, the user needs to view the scene from different perspectives to gather more visual information. For example, in a grasping task the human may need to see the object from the side view to ensure the robot gripper is aligned. Therefore, we allow the user to "inspect" the 3D local scene from any angle and distance (Fig. 2(c)). This is triggered by a gripper button on controllers and shows only the point cloud from the local camera. The system tracks the user's head movement via the HMD and orientates the point cloud accordingly. When the user moves towards the point cloud, the view is zoomed in to present a closer view. The speed of the view changes is heuristically chosen to give the user a natural and comfortable view experience. The user is able to switch back to the PCP when he/she is done with the inspection.

III. Experiment Setup

In the system, the robot was remotely controlled by an operator via a network. The network was composed of two nodes. One node was running the Unity in a Windows10 machine to collect the operator's command and another node was running on ROS (Kinetic) in a Ubuntu 16.04 machine to receive operator's command and control UR5 using URscript language. Two nodes were wirelessly connected via WIFI. The communication frequency was 125Hz and implemented using ROS Unity bridge [x].

The UR5 was working in the joint servoing mode. Its desired joint angles were computed incrementally. The desired angles were equal to the current measured angles plus the desired joint angles rate multiplied by the control period. The current measured angles were obtained online, and the desired joint angles rate was computed from the readout of HTC Vive controller (the desired twist motion of robot) and inverse kinematics model of robot. The HTC Vive controller firstly needed to be calibrated. Via the controller, the operator can give the desired linear and rotation velocity to carry out the given task according to his/her intuitive observation in the virtual scene. The desired twist command was given according to the local coordinate frame of the robot end-effector and this can avoid the requirement that operator remaps the motion command back to global frame in his memory. Other parameters which is necessary for servoing joints of UR5 were manually tuned to guarantee the smooth movements of the robot.

Participants were asked to perform a manipulation task multiple times using the proposed working modes. The task included picking up two objects (a ball and a cube) from an opaque box and putting them on a table. The objects were occluded by the box and the participants had to carried out the task based on their observations in different modes, as shown in Fig. 2.

[x]https://github.com/siemens/ros-shar

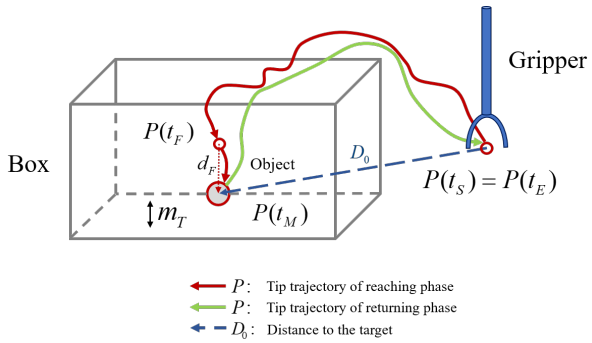


Fig. 3: Objective metrics for evaluation of manipulation task.

For data analysis, we recorded the time the users started to move the robot t_S, the time they reached the target t_M (successful touch), and the time when the manipulation was completed successfully t_E based on the kinematics data of the robot and the environmental parameters. We also recorded the time t_F the users first attempted to grasp an object in each trial, i.e. their first attempt without any prior or feedback from the environment. As shown in Fig. 3, we computed the trajectory of the robot gripper tip based on its parameters and the kinematics data: $P(t)$, $t_S < t < t_E$. The number of attempts the users tried to manipulate the target object m_T was also measured in each trial.

We propose the following four objective metrics to evaluate the performance of teleoperation with different working modes : efficiency/speed to reach the target (E_{reach}, S_{reach}), efficiency/speed to complete a fetch task (E_{fetch}, S_{fetch}), number of attempts of grasp(m_T), first grasp distance(d_F).

A. Efficiency/Speed to Reach the Target (E_{reach}/S_{reach})

In experiments, the users may tend to get as close to the objects as possible before grasping them. Smoothly approaching and touching the target in the reaching phase was the key to successful grasps. We utilized the speed and efficiency to describe teleoperation's performance in the reaching phase. E_{reach} and S_{reach} are defined as:

$$E_{reach} = \frac{\|P(t_M) - P(t_S)\|}{t_M - t_S}, S_{reach} = \frac{\int_{t_S}^{t_M} \sqrt{1 + \dot{P}(t)^2}}{t_M - t_S} \quad (2)$$

The norm of the vector from the robot gripper to the target was used as a measure of distance in tasks. The length of the trajectory in the reaching phase was computed by integrating the speed measurement.

B. Efficiency/Speed to Complete a Fetch Task (E_{fetch}/S_{fetch})

Similarly, we also described teleoperation's performance in the whole fetch task (reach, grasp and return) by speed(S_{fetch}) and efficiency(E_{fetch}), which are defined as:

$$\begin{aligned} E_{fetch} &= \frac{\|P(t_M) - P(t_S)\| + \|P(t_E) - P(t_M)\|}{t_E - t_S} \\ S_{fetch} &= \frac{\int_{t_S}^{t_E} \sqrt{1 + \dot{P}(t)^2}}{t_E - t_S} \end{aligned} \quad (3)$$

C. Attempts of grasp(m_T)

The operators may try different poses of robot for grasping after they have located the target. We counted the number of attempts in grasping (m_T) until they successfully picked the target up . m_T is a direct indicator of the performance in grasping targets; the more attempts users made to grasp the target, the less the efficiency they demonstrated in the grasping phase.

D. First Grasp Distance(d_F)

With the assistance of different AR interfaces, the users established their sense of the environment in the occluded view so that they can approach and grasp objects. d_F, defined as the distance to the target in the first manipulation attempt, is a indicator of the accuracy of the sense of environment:

$$d_F = \|P(t_M) - P(t_F)\| \tag{4}$$

In the first attempt, the users usually believed that robot have reached the target and could start grasping it with the pose, without feedback of the real pose of the gripper. Smaller d_F indicates a higher accuracy of the sense of environment established by AR.

The whole experimental procedure was defined as follows:

1) Complete the consent form (user study only) and pre-experiment survey.
2) Complete the training of picking up objects and putting them on the other side of table, in three working modes (PIP, PCP, PCI) and repeated multiple times.
3) Pick up two objects (a cube and a ball) from a box with occluded view and put them on a table, with PIP,PCP and PCI in a random order.
4) Complete the NASA Task Load Index [20] questionnaire (TLX) and answer several supplementary questions after each trial (three times).
5) Complete the post-experiment survey for the task.
6) Conduct an informal interview.

The post-experiment survey includes: i) self-reporting ratings on intuitiveness, ease of use and future prospect for three interfaces(1-7), ii) preference of the three teleoperation interfaces (0-1) and the reasons. The supplementary questions in the forth stage include: i) whether the FOV, smoothness, latency, quality of the rendered point cloud, accuracy of the virtual overlay or any other factors limit the current application (Y/N), and which is the most limiting factor, ii) whether the inspection mode helps in the experiment (Y/N) (PCI mode only).

All the related data was recorded in every trials, including the objective data, live video, TLX questionnaire and post-experiment survey. All data were timestamped in millisecond.

IV. User Study

A. Subjects and experiments

We conducted a user study with a total of 20 subjects from the local community (14 male, 6 female, ages 20 to 40), all have their inform consent forms signed. Three of them had experience with VR, but none of them had any experience of robotic teleoperation. To reduce the inconsistency of users'

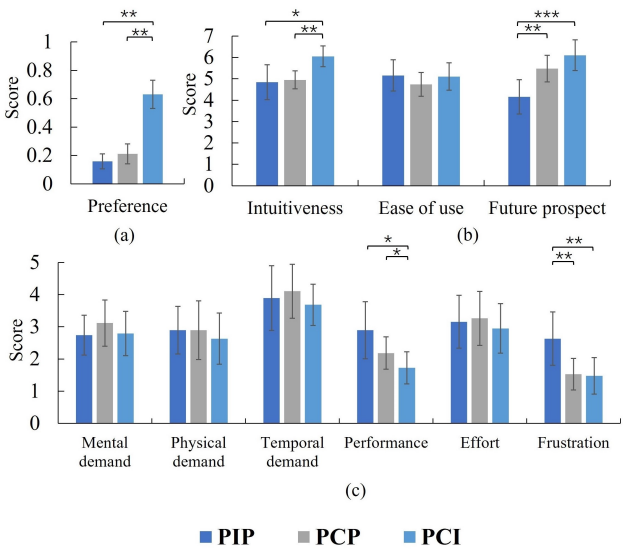


Fig. 4: Ratings for subjective metrics. (a) Preference. (b) Intuitiveness, Ease of use and Future prospect. (c) Results of TLX questionnaire. ("*" indicates a significant difference, p <0.05; "**" indicates that p <0.01; "***" indicates that p <0.001).

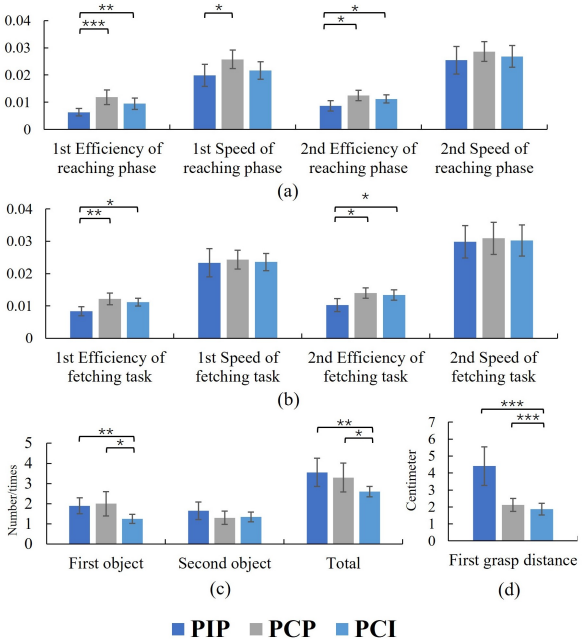


Fig. 5: Results for objective metrics. (a) Efficiency/speed in reaching phase. (b) Efficiency/speed in fetching phase. (c) Attempts of grasp. (d) First grasp distance.

understanding of questionnaires, all the subjects had technical background.

Before the manipulation task, each participant was given a thorough explanation on the experiment setup, the robot, and the procedure. In our experiment, we only displayed the virtual robot gripper fingers as they were the most important part for the manipulation in the experiment. Extra markers were also used in the scene to increase the number of feature points for alignment.

All the subjects completed repetitions of three interfaces, and the order of the three trials was generated randomly to minimize the effect of prior knowledge. Among different trials in a task, the pose of objects was reset manually by the assistants. We timed the duration of each trial and asked the participants to complete the TLX questionnaire and the post-experiment survey about their impression of each interface. In the last stage, we recorded users' experience in trials and their suggestions for the whole system in the interview.

B. Results and analysis

1) Subjective results: In questionnaire and interview, participants rated their experiences of the experiment in several aspects. We found a significant effect of our interface design on users' favor working in occluded environment with robot.

As shown in Fig. 4(a), users were more inclined to utilize PCI mode with the preference 0.632 higher than that of the PIP mode ($M = 0.158$, $p = 2.05 \times 10^{-3}$) and the PCP mode ($M = 0.211$, $p = 7.60 \times 10^{-3}$). The most stated reasons for the tendencies were: "the mode shows a fascinating 'perspective' effect and shows the local pose of objects precisely, PIP always misses the depth information from the overlook perspective (the same reason for PCP)". Other reasons for PCI are: "allowing users to observe locally at close range" and "enables the users to concentrate on the task", etc. Reasons for those choosing PIP were consistent: "a more familiar way to me and easy to use", although many of them were quite interested to PCP and PCI in interview. It is worth mentioning that 4 subjects preferred PCP and they thought that PCP put all the information together which was more promising.

As shown in Fig. 4(b), the average rating for intuitiveness in PCI mode was 6.05, which was significantly higher compared to that of the PCP mode ($M = 4.95$, $p = 3.21 \times 10^{-3}$) and the PIP mode ($M = 4.84$, $p = 3.43 \times 10^{-2}$). Meanwhile, the users thought that the PCI mode ($M = 6.11$, $p = 6.04 \times 10^{-4}$)) and the PCP mode ($M = 5.47$, $p = 3.66 \times 10^{-3}$) had better future prospect compared to that of the PIP mode ($M = 4.16$) significantly. For the ease of use, the average rating for the three modes were relatively close and the t-tests among them were not statistically significant: 5.16 for PIP mode, 5.11 for PCI mode and 4.74 for PCP mode. Results of TLX questionnaire is shown in Fig. 4(c), the average ratings for PIP mode were: 2.7 (mental demand), 2.9 (physical demand), 3.9 (temporal demand), 2.9 (performance), 3.1 (effort) and 2.6 (frustration). For PCP mode, their average ratings were 3.1, 2.9, 4.1, 2.2, 3.3 and 1.5. For PCI mode, their average ratings were 2.8, 2.6, 3.7, 1.7, 2.9 and 1.4. It is interesting to notice that "ease of use" is similar but "intuitiveness" is different across the modes. It suggests that a part of users felt PCI helped them to understand the scene better but was less helpful in manipulation. They considered that PIP is a more familiar mode and they could do a good job in this mode. This can be caused by the fact that the depth information provided by the point cloud is useful for understanding, but it's accuracy is not good enough for manipulation. This could also explain the insignificance of the TLX results in mental/physical/temporal loads and efforts across the different modes.

As for the limitation of the system, the loss of depth information ranked first (12 subjects) in trials using PIP interface. Quality of the rendered point cloud was cited the most (11&9 subjects) in trials using PCP and PCI interfaces, which was consistent with our assumption in the objective analysis. The majority of the users considered the inspection function was helpful in the experiments, and they thought that the mode provided more detailed perspectives which helped grasping objects firmly, allowed them to focus on the task itself.

2) Objective results: To understand the results in subjective metrics in depth, we have analyzed the objective data by the detailed metrics. Normality tests were performed before the analysis of whether the data satisfy the normal distribution assumption. T-tests were conducted for the metrics to justify the statistical differences between three AR interfaces. A p-value$<$0.05 was considered significant.

Fig. 5(a) shows that when approaching the first object, the efficiency in reaching phase for PCP ($M = 0.0118$) increased significantly ($p = 2.56 \times 10^{-4}$) by 87% , for PCI ($p = 6.62 \times 10^{-3}$) increased by 51% ($M = 0.0095$), compared to the traditional PIP mode ($M = 0.0063$). For the reaching speed, users moved the robot with 2.57cm/s in PCP mode, which was significantly ($p = 1.83 \times 10^{-2}$) higher than the velocity in PIP mode ($M = 1.97 cm/s$). PIP did not show statistical difference with PCI in reaching speed. When reaching the second object, the efficiency for PIP ($M = 0.0082$) was significantly lower than the PCP ($M = 0.0125$, $p = 1.24 \times 10^{-2}$) and PCI mode ($M = 0.0110$, $p = 1.41 \times 10^{-2}$), the results of reaching speed in the process were not statistically difference.

Fig. 5(b) shows the results of the efficiency and speed in the whole tasks. When fetching the two objects, efficiency in PCP($M_1 = 0.0121$, $p_1 = 7.24 \times 10^{-3}$, $M_2 = 0.0140$, $p_2 = 2.96 \times 10^{-2}$) and PCI($M_1 = 0.0112$, $p_1 = 1.48 \times 10^{-2}$, $M_2 = 0.0134$, $p_2 = 1.12 \times 10^{-2}$) mode were significantly higher than that of PIP($M_1 = 0.0083$, $M_2 = 0.0102$). The results of speed in the whole tasks were not statistically difference.

As shown in Fig. 5(c), the attempts of grasping the first object significantly ($p = 2.67 \times 10^{-3}$) reduced by 34% and 37.5%, from 1.9 times in PIP and 2 times in PCP to 1.25 in PCI mode. When trying to picking up the second object, the number of attempts were not significant. In summary, the total number of grasping trials show similar tendencies as the number of trials to pick up the first object. ($M_{PIP} = 3.55 times$, $M_{PCP} = 3.3 times$, $M_{PCI} = 2.6 times$)

The results of d_F, the distance to the target at the first grasp attempt, are shown in Fig. 5d. The mean d_F reduced from 4.41cm in PIP mode to 2.12cm in PCP mode ($p = 1.32 \times 10^{-4}$), and 1.87cm in PCI mode ($p = 2.85 \times 10^{-5}$) by 52% and 58% significantly. PCP and PCI did not show statistical difference in t-test.

In summary, PCP and PCI showed higher efficiency than PIP in the reaching stage and the whole task, they also outperformed PIP significantly in the smaller first grasp distance, which indicates a better telepresence. According to the efficiency in grasping attempts, PCI showed significantly better performance than PCP and PIP.

It is worth mention that PCP and PCI performed relatively better in reaching phase than in the whole task compared to PIP. One potential reason may be that the users spent more time in two modes than in PIP in adjusting pose to grasp firmly after they have reached the target, due to the missing part of the rendered point cloud. The problem will be solved in future by mounting another local camera on the other side of the gripper to complete the point cloud.

V. Conclusion and future work

In this paper, we develop an intuitive augmented reality interface based on multi-view fusion and dynamic calibration approach to endow people with immersive experience. We conducted a user study with 20 inexperienced users and collected experimental data. The results showed that our proposed approach can benefit users by improving efficiency in manipulation task, especially in the reaching stage comparing with traditional PIP approach.

In the interview, many users gave many perspicacious suggestions. They are summarized as following:

- Reducing the latency of perception and control
- Adjusting the flexibility of inspection mode to make sensitive users more comfortable.
- Adapting the transparency of the point cloud to make all the objects visible in different scenarios.
- Merging point clouds locally with two cameras mounted on the end-effector for better quality.
- Identifying operator's intention with machine learning algorithms to "intelligently" shared control the robot instead of operator's fully direct control.

These suggestions will be taken into consideration in our future works.

VI. Acknowledgement

We acknowledge Nicholas Gerard Timmons, Hao Qin for their supports in developing the system. Qiang Li is supported by the "DEXMAN" project (Project number: 410916101) funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation).

References

[1] T. Klamt, D. Rodriguez, M. Schwarz, C. Lenz, D. Pavlichenko, D. Droeschel, and S. Behnke, "Supervised autonomous locomotion and manipulation for disaster response with a centaur-like robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.

[2] R. Montero, J. G. Victores, S. Martinez, A. Jardón, and C. Balaguer, "Past, present and future of robotic tunnel inspection," *Automation in Construction*, vol. 59, pp. 99–112, 2015.

[3] L. Qian, A. Deguet, Z. Wang, Y.-H. Liu, and P. Kazanzides, "Augmented reality assisted instrument insertion and tool manipulation for the first assistant in robotic surgery," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5173–5179.

[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] 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.

[6] D. Ni, A. Nee, S. Ong, H. Li, C. Zhu, and A. Song, "Point cloud augmented virtual reality environment with haptic constraints for teleoperation," *Transactions of the Institute of Measurement and Control*, vol. 40, no. 15, pp. 4091–4104, 2018.

[7] 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.

[8] 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.

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

[10] 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.

[11] G. J. Anderson and C. S. Marshall, "Augmented and virtual reality picture-in-picture," Oct. 4 2018, uS Patent App. 15/476,119.

[12] Y.-T. Lin, Y.-C. Liao, S.-Y. Teng, Y.-J. Chung, L. Chan, and B.-Y. Chen, "Outside-in: Visualizing out-of-sight regions-of-interest in a 360 video using spatial picture-in-picture previews," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 2017, pp. 255–265.

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

[14] S. Kohn, A. Blank, D. Puljiz, L. Zenkel, O. Bieber, B. Hein, and J. Franke, "Towards a real-time environment reconstruction for vr-based teleoperation through model segmentation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[15] T. Dardona, S. Eslamian, L. A. Reisner, and A. Pandya, "Remote presence: development and usability evaluation of a head-mounted display for camera control on the da vinci surgical system," *Robotics*, vol. 8, no. 2, p. 31, 2019.

[16] K. Theofilis, J. Orlosky, Y. Nagai, and K. Kiyokawa, "Panoramic view reconstruction for stereoscopic teleoperation of a humanoid robot," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 242–248.

[17] B. Omarali, B. Denoun, K. Althoefer, L. Jamone, M. Valle, and I. Farkhatdinov, "Virtual reality based telerobotics framework with depth cameras," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2020, pp. 1217–1222.

[18] Y. Chen, B. Zhang, J. Zhou, and K. Wang, "Real-time 3d unstructured environment reconstruction utilizing vr and kinect-based immersive teleoperation for agricultural field robots," *Computers and Electronics in Agriculture*, vol. 175, p. 105579, 2020.

[19] B. Huang, N. G. Timmons, and Q. Li, "Augmented reality with multi-view merging for robot teleoperation," in *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 260–262.

[20] S. G. Hart, "Nasa-task load index (nasa-tlx); 20 years later," in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 50, no. 9. Sage publications Sage CA: Los Angeles, CA, 2006, pp. 904–908.