# Instinctive Robot Control via Hololens 2

Jonathan Becker
jonbecke@student.ethz.ch

Michael Baumgartner
michbaum@student.ethz.ch

Ivan Alberico
ialberico@student.ethz.ch

Seif Ismail
ismails@student.ethz.ch

## Abstract

*Mixed reality applications are becoming increasingly widespread in numerous areas, ranging from the medical surgical domain to entertainment, design and more. At the same time autonomous robots are demonstrating incredible potential in the domain of autonomous inspection tasks in hazardous environments and are already omnipresent in the assembly lines around the world. This paper seeks to combine these two promising research avenues through the development of a novel, instinctive mixed reality teleoperation interface on a Microsoft HoloLens 2. This interface allows users to remotely act on a physical scene through a robotic arm by interacting with a digital twin of the scene and robot in the mixed reality space. This in turn opens up unique teleoperation applications presently not feasible due to their non-standardised nature. A summative users study shows that the proposed method enables untrained users to perform basic pick and place tasks. The interaction further achieves a very good score on the standardised user experience questionnaire.*

## 1. Introduction

Recent advances in the field of mixed reality have made the devices lighter, cheaper and more accurate, leading to an increasing acceptance of mixed reality as a new useful human-machine interface [15]. The technology enables new methods of interacting with 3D scenes and objects, which has particularly interesting implication for the remote control of robots and robotic arms. Previous works in this field have shown that mixed reality devices like Microsoft's HoloLens 2 or smartphones can be used to program standardised tasks. Those include picking up an object and placing it by either telling the robot where the object is and where to place it, or by giving the robot a series of waypoints [7, 9, 12]. These approaches typically still require manual programming or are limited in regards to non-standardised tasks, like closing a valve, opening a control cabinet or flipping a lever.

In this work, we aim to combine the intelligence of humans with the utility of robotic arms. For this, we present a novel and instinctive interface for the remote control of a robotic arm, making it behave as an extension of the user's arm. This alleviates the need for tedious programming and opens up a wide range of applications. We already see examples of high-impact applications that can potentially be extended by such an interface in the work field today. An example of which is Boston Dynamic's *Spot*, which is being used in the visual inspection of hydro-power plants. In these scenarios, robotic agents are often acting autonomously and are only supervised remotely. Equipping inspection robots with robotic arms would extend the work that could be performed remotely, ultimately reducing the cost of maintenance and making it safer at the same time. With the control method developed in this work, a human operator can use a robotic arm to perform a variety of maintenance tasks over a distance without any direct line of sight to the robot itself. To evaluate our control interface, we implemented it on an off-the-shelf *WidowX 250s 6 DOF* robotic arm in conjunction with a HoloLens 2 device and used a *Intel RealSense D435i* RGB-D camera and ArUco markers to complete a cube-stacking challenge designed to assess its precision and instinctiveness. A demonstration of the final prototype can be found on YouTube [1] and the source code is freely accessible [2].

### 1.1. Related Work

The remote control of robotic arms is a popular and ongoing research topic, especially in the field of medical robots. As surgeons typically require haptic feedback, they use physical input devices [22] to control the robotic arms [10]. These devices provide very accurate inputs, but they are expensive and usually only have a small range of motion specific to surgical applications or tasks like welding [21].

[1] https://youtu.be/YiZyG_5g66w
[2] https://gitlab.ethz.ch/mr-instinctive-robot/mr-instinctive-robot-control

To extend the range of motion, researchers evaluated different input modalities like sensor-packed gloves [19], colour-coded gloves [20] that were tracked via external cameras or pointing devices with reflective markers [13]. Other common but less direct input devices include joysticks, dials or robot replicas, with the lowest level of input being direct API calls [16].

Augmented reality (AR) devices have previously been used to display the state or planned course of action of a robot. Researchers also combined AR with a pointing device to program a robot in his physical space by defining goal positions, waypoints and obstacles [13, 18].

During the course of this project, the company *Extend Robotics* presented a teleoperation application featuring a digital twin of a robotic arm in virtual reality (VR) [1]. In their method, the user controls the robot by dragging specific parts of the digital twin with Oculus Quest controllers. This movement is replicated by the physical robot. Notably, they also draw a point cloud of the robot's surroundings in the VR space, allowing for spatial awareness of the user and unconstrained interaction with the robot's environment.

The communication between the robot and the MR device is realised in most works following the architecture proposed in [14], by using a server, sometimes called broker, in between the MR input device and a computer connected to the robot. The computer runs code written for the Robot Operating System (ROS) to control the robot and potentially other things which are then sent to the MR device via a TCP or UDP connection. Unity itself released an open source framework for this application [4].

Our approach differs from prior work mainly in its input modality and its strong focus on the instinctiveness of the control interface. Our control interface is, to the best of our knowledge, the first of its kind that solely relies on hand pose estimations as its input whilst still achieving high accuracy and low latency pose tracking of the robotic end-effector.

### 1.2. Contributions and Challenges

In this work, we leverage the benefits of MR to enable the intuitive remote control of a robotic arm by controlling a digital twin as if it were an extension of the user's arm. This removes the need for programming and allows a non-trained operator to control the robot instinctively. We achieve accurate, low latency pose tracking of the user's hands without using an additional input device. We further enable users to grasp objects by tracking individual finger joints and introduce a series of design elements aimed at improving the overall user experience. Finally, we demonstrate the effectiveness and intuitiveness of the approach by performing a user study. To achieve this, we had to overcome several challenges including:

- Setting up a communication framework between the HoloLens 2, a server and the robot, as well as documenting the steps for reproducibility of the open source framework.
- Identifying and defining new interfaces with the low- and high level controllers that came with the robotic arm.
- Accurately tracking cubes in the robots environment and correctly positioning them with regards to the robot in the MR space of the user.
- Encapsulating everything in a user friendly Unity application with easy to understand visual feedback.

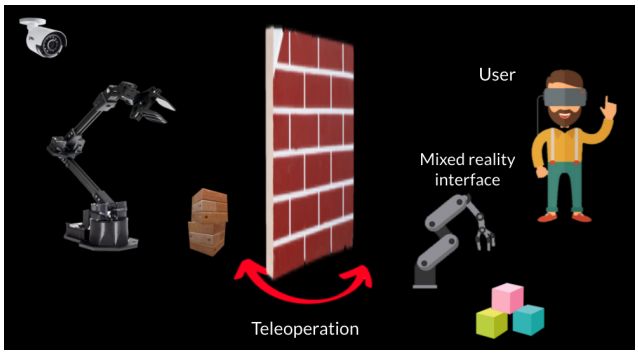


Figure 1. Abstraction of the problem setup: User interacts with a digital twin of the robot and the scene consisting of three cubes. The robot is teleoperated and follows the user inputs without direct line of sight.

## 2. Problem Definition

The goal of the project was to develop an intuitive MR interface on a Microsoft HoloLens 2, in which any non-trained operator could remotely control a robot arm and perform a basic assembly task using hand tracking. To achieve a measurable outcome, the assembly task was further refined and limited to the final challenge of stacking three cubes of known size on top of each other. A schematic of the problem setup can be seen in Fig. 1. To meet the intuitiveness requirement, we further required that any action intended by the operator should be accurately and responsively followed by the robot in the physical environment and that any change in the physical environment in turn should be mapped into the MR environment. Additionally, following Nielsen's heuristics for user interfaces [17], the state of the system should be clearly visible at all times, the interaction should be efficient, and enough help and documentation should be provided for an inexperienced user. To achieve the defined task, the following hardware was used:

- Intel RealSense D435i: RGB-D camera for scene reconstruction.

- WidowX 250s Robot Arm 6 DOF: for interaction with the physical environment.
- HoloLens 2: for visualisation and interaction in the MR environment.
- Plastic cubes of 7 cm side length
- A custom 3D printed robot end effector (sometimes referred to as "gripper").

## 3. Method/Implementation

To teleoperate the robot arm and manipulate its surroundings, three major challenges were identified and individually addressed. The biggest challenge was to make the robot follow the user's hands smoothly, whilst remaining within reachable bounds. This included the adaption of novel robot controllers and the development of an intuitive method for interacting with the robot in mixed reality.

Closely linked to that was the second challenge of designing an adequate input modality based on the tracked hands and finger joints. This involved identifying a comfortable and logical transformation between the hand pose and end effector pose, but also coping with the inherent properties of the hand tracking interface offered by the HoloLens 2.

Finally, to enable interaction with the surrounding, or in this case the cubes, a robust and accurate estimate of the cube poses relative to the robot had to be acquired. The two key flows of information and an overview of the entire system are shown in Fig. 2 and are further detailed in the following sections.

### 3.1. Utilising Hand Poses as Control Inputs

One of the key defining factors in the design for an instinctive robot controller lies in the choice of the input modality. Whilst a multitude of previous papers [13, 19, 20, 22] showed great promise in the application of mixed reality or augmented reality devices for teleoperation tasks, virtually all of them relied on some sort of a controller or auxiliary device for user input.

In this work, the actual user hand pose is chosen as the input modality and hand gestures as command inputs, with the hypothesis that such a choice would amplify the user ownership [6] and thus the instinctiveness of the approach. This however necessitates an accurate hand tracking module as well as an intuitive input pose. Estimating accurate hand poses is an ongoing research field, having to deal with broad generalisation challenges regarding different hand sizes, skin tones and other irregularities in the user's hands. Our approach utilises the hand tracking capabilities of the HoloLens 2, which leverages a depth camera for accurate pose estimations of 25 predefined joint poses in the user's hand (Fig. 3, left). One can then interface these 25 joint poses directly in Unity. To allow for a hand size agnostic pose estimation, we chose the input position relative to two of those hand joints, namely the thumb proximal and the index knuckle joint.

$$input\_pos = thumb\_pos + \frac{index\_pos - thumb\_pos}{2} \quad (1)$$

This allows us to always co-locate the virtual robot end effector model and the hand in the same way, irrespective of hand geometry. For demonstration purposes we consider only *right handed* users that control the robot with their dominant hand, whereas the left hand is used for auxiliary functions as described in Sec. 3.6. Efforts have also been made to align the gripper orientation with the plane spanned by the thumb metacarpal, proximal and the index knuckle joint to allow the user to grip objects in the most natural way possible (Fig. 3, right).

The main hand gesture used as an input is the act of bringing one's index tip and thumb tip together to close the robot's gripper. Two thresholds are incorporated to send closing signals when the gripper is still open and opening signals when the gripper is closed, respectively. This is done to diminish the effects of inaccuracies in the pose estimation of the finger joints, which could lead to involuntary control inputs and a failed task.

### 3.2. Translating Inputs into Robot Motion

The main goal of the teleoperation controller is to make the robot move as if it was an extension of the user's arm, ideally taking the place of their hand in the mixed reality space. To achieve this, we identified two promising control methods that were to some degree already implemented on the robot used for this project and evaluated both of them.

The first option is based on the popular *MoveIt* control library [2], which is commonly used for robot arms. It involves using an inverse kinematics solver to plan a joint trajectory for the robot arm to make it move to a desired pose. This approach is however limited due to its computational expense and hence introduces a significant delay between the time that the desired pose is sent and the time that the robot actually moves there. Additionally, the solver requires the exact start and end positions of the robot arm, hence the robot arm needs to finish the execution of the previous trajectory before a new path can be planned and executed. Sending continuously updated hand poses to this controller thus leads to a counter-intuitive stop-and-go behaviour of the robot. As the project put strong emphasis on the intuitiveness of the human-robot interaction, this control approach was abandoned.

The second controller that was evaluated speeds up computation by only solving the inverse kinematics iteratively

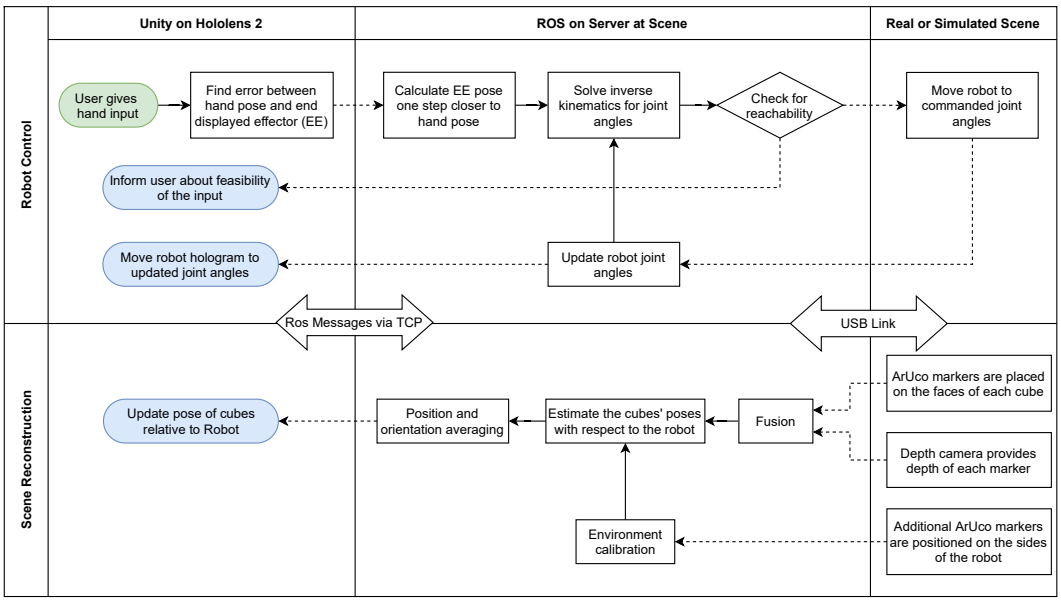


Figure 2. Flowchart giving an overview of the building blocks making up the application and the flow of information between them. Green blocks indicate inputs, blue blocks indicate outputs.

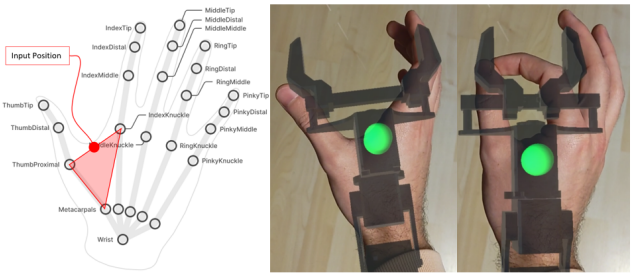


Figure 3. Visualisation of the 25 estimated joint poses in the HoloLens 2 interface with our control input position (left image) as well as the actual co-location of gripper and the user's hand in the application (right images). The rotation of the gripper was calibrated to align with the plane spanned by the thumb metacarpal, proximal and the index knuckle joint.

for small steps of the end effector towards the goal pose. This approach allows for a continuous movement of the end effector towards the hand pose. Employing this control method results in a much more satisfactory user experience, as the robot follows hand motions immediately, accurately and with little delay. However, due to the iterative computation of only locally consistent solutions, using this new approach introduces limitations to the operational space of the robot. The inverse kinematics solver regularly fails to find solutions for seemingly reachable positions, whenever the robot arm is near a singularity. This happens most notably when some joints are close to their limits. To address this limitation, a *home position* with desirable joint states far away from singularities was introduced. The robot is initialised to this position and can be reset at any time. Additionally, a glowing green box indicating the working volume was added in Unity, which limits the area in which the robot tries to follow the user's hand to account for the locality of the controller. This ensures that the robot is able to follow the user inputs at virtually all times when inside the volume. Finally, to regain the full range of motion of the robot, a feature to move the current working volume was added. For this, an ephemeral working volume is displayed around the robot at the position closest to the user's right hand as soon as he leaves the current working volume. The working volume is then moved to the displayed position as soon as the user holds together his right thumb and index finger for longer than 0.75s. This approach is based on a workflow where a user picks up an object in one working volume, then rotates the robot around the waist and finally places the object in another working volume. As further detailed in Sec. 4.1, this method of moving the working volume was perceived as obstructive and counter-intuitive by some users, mostly because it triggered on accident. A possible solution to this would be the incorporation of voice commands as further discussed in Sec. 5.2.

### 3.3. Tuning of the Controller

The iterative controller assigns the same constant execution time to each control input irrespective of the step size given. Thus, to increase the smoothness of the controller, the size of each step is adapted proportionally to the error between robot and hand, increasing as the error grows but remaining below a defined maximum value. This leads to

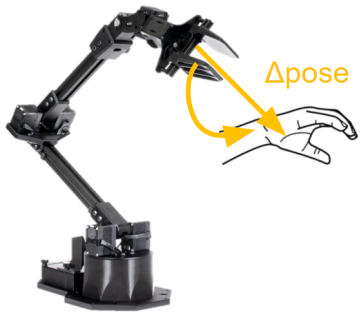


Figure 4. The picture is a schematic representation of how the controller works: the control input to the robot is computed based on the difference between the end effector pose and the hand pose.

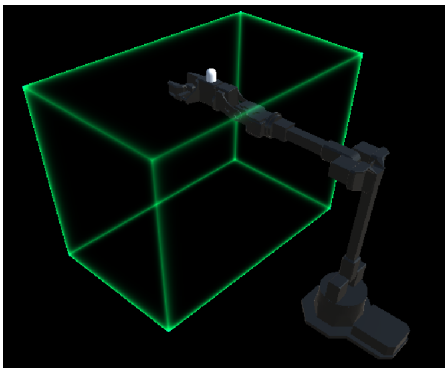

Figure 5. Representation of the operational working volume of the robot in Unity.

a saturated proportional control. For small gains, the end effector moves only slowly towards the hand pose and is no longer perceived as an extension of the user's arm, whereas for large gains the end effector overshoots the desired pose and starts to oscillate about it. Satisfactory results were obtained with gains found through iterative tuning. Additionally, a *deadzone* around the goal pose was added, which ignores errors close to zero. This mitigates noise in both the hand and robot pose estimation and thus reduces jitter.

### 3.4. Feedback of the System State

Feedback of the system's state is provided in multiple ways. Besides the objects in the 3D scene moving according to the physical scene, the green glowing box shown in Fig. 5 indicates whether the user's hand is inside the working volume of the robot or not. Additional feedback to the user is provided through a lamp on top of the holographic robot end effector, shown in Fig. 6. The colour of the lamp signals whether the robot is sleeping (black), awake (white), able to follow the user's hand (green) or in a configuration where it is not able to reach the current hand pose (red).

Figure 6. (left) The green light shows that the user input is a feasible pose for the robot. (right) The red light shows that the robot is not able to reach the pose provided by the user.

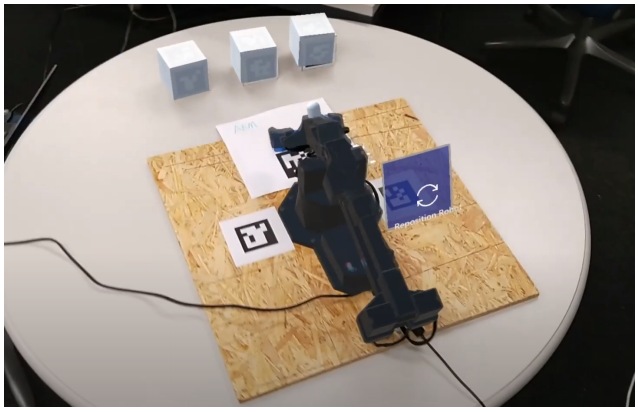

Figure 7. Mixed reality scene of the HoloLens 2: The hologram of the robot has manually been positioned on top of the physical robot. Note that the holographic cubes are correctly overlaid on top of the actual cubes, indicating that the scene is accurately reconstructed.

### 3.5. Cube Pose Estimation

To remotely interact with the scene around the robot, the elements of interest need to be reconstructed in the mixed reality interface. For this, they have to be detected and tracked in the physical scene. Object detection itself is still one of the main open research fields in computer vision and it comes with numerous challenges, like dealing with occlusions, illumination changes, viewpoint variations and cluttered or textured backgrounds. We reduced the complexity of this task based on the problem definition developed in Sec. 2. We employ a *marker based* detection approach leveraging ArUco markers, since the size and geometry of the interactable objects in the scene are assumed to be known. This allows for a robust and rapid implementation of the cube pose estimation module, with which the control interface can be tested. The ArUco markers are placed on all faces of each cube, which makes the detection robust to occlusions. In addition to that, ArUco marker detection is computationally inexpensive, allowing for real-time streaming of the cube poses to the mixed real-

ity scene (Fig. 7). For further refinement, the information provided by the depth camera is utilised. It provides an estimate for the depth of each marker with millimetre accuracy. This is then fused with the ArUco pose estimation module in OpenCV [3], which allows to extract both position and orientation of the centre of each detected marker in the frame. Two steps are performed to convert the multiple marker poses into the pose of the cube centre relative to the camera. The output of the OpenCV ArUco pose estimation module is the relative pose of the reference system of each detected marker with respect to the camera. Since the frames of the different faces are not aligned, their poses are first expressed with respect to a predefined dominant face. Then, to obtain the cube centre, the position of each marker is translated by half the cube size along the normal of the face. From Fig. 8 it becomes clear that this always corresponds to the z axis of the ArUco marker. Finally, the cube centers from multiple markers are averaged and combined with knowledge of the previous pose of the cube, to find its updated pose. This estimated pose is still expressed in the camera reference frame and needs to be transformed into the robot base frame. To achieve this, two markers were placed at the sides of the physical robot for calibration. The relative poses of these markers are found analogous to above. Knowing the static transformation between these markers and the robot allows for the expression of the cubes' poses in relation to the robot frame.

The cube pose estimation module is implemented as a ROS node. It publishes information on whether the specific cube has been detected and its relative pose with respect to the robot. A TCP connection between the ROS node and Unity is established [4], in order to stream and update the cube poses in the mixed reality scene at each frame. In Unity the cubes are spawned as child objects of the robot base frame, so that whenever the robot is moved in the mixed reality space (i.e., due to repositioning), the cubes move accordingly. One final challenge that had to be overcome in order to guarantee a better user experience was to attenuate the jittery behaviour encountered while visualising the holograms of the cubes in Unity. This jitter is caused by irreducible mismatches among the poses estimated from the different faces of the cube. Whilst the effect on the position is diminished by averaging the position vectors of each detected face, problems arise as far as orientation is concerned. Orientations are encoded as unit quaternions in Unity, and unit quaternion averaging is still an open research topic. We implemented an efficient solution following [11], which solves the averaging as an optimisation problem based on its eigenvector decomposition, as shown in the following:

$$M = \sum_{i=1}^{n} w_i \; q_i \; q_i^T \tag{2}$$

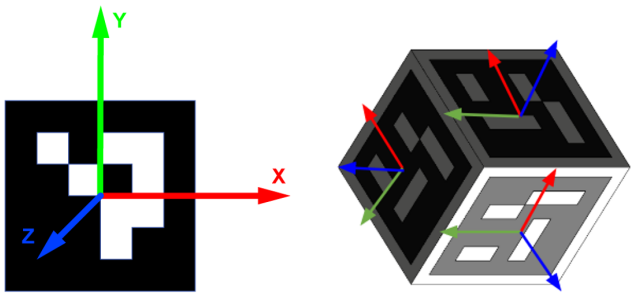


Figure 8. (a) Reference frame of a single ArUco marker. The z-axis is always directed along the normal of each face of the cube. (b) Configuration of the cube for 3 simultaneously detected faces. In this case, the position of the centre of the cube is estimated by averaging over all detected faces.

$$\bar{q} = \underset{q \in \mathcal{S}^3}{\operatorname{argmax}} \, q^T M q \tag{3}$$

where $w_i$ refers to the weight associated to each quaternion $q_i$ in the averaging process. The averaged quaternion is the eigenvector of M corresponding to the maximum eigenvalue, and it has unit norm by construction.

### 3.6. Additional Features

To improve the user experience, an extensive tutorial has been added. The tutorial provides a step-by-step explanation on how to interact with the robot and how the different functionalities work and can be used. Among the functionalities that have been implemented, there is a *Repositioning* button, that allows the user to freely move the robot and the work space wherever they prefer. In addition to that, we also implemented a hand menu that appears whenever the user looks at the palm of his left hand. The menu contains the following commands: *Start Following Hand*, *Reset Position*, *Go Sleep* and *Tutorial*. The first one allows the start of the hand tracking and path following modules, so that the robot starts following the user's hand only when intended. This is a crucial safety functionality, since it avoids involuntary robot movements right after the application starts. As previously stated, the *Reset Position* button brings the robot to a the *home position* and the *Go Sleep* button returns the robot to its sleeping configuration. Finally the *Tutorial* button brings up the tutorial.

## 4. Evaluation

A summative user study was conducted with the final prototype to evaluate its quality in terms of user experience and especially intuitiveness, effectiveness and ease of use. Moreover, multiple unrelated, small-sample, formative user studies have been conducted throughout the iterative design process of the software. These were aimed at obtaining an

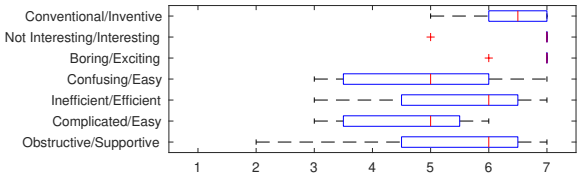


Figure 9. Survey Results: Box plots summarising the answers of eight participants to the UEQ Questions, where 1 is the lowest and 7 is the highest score. The red line indicates the median of the data. The box indicates the 25 to 75 percentile or Inter Quartile Range (IQR). Points outside of $1.5 \cdot$ IQR are marked with a cross and are considered outliers.

outside view on what we thought to be more or less intuitive, allowing us to refine our application design, user feedback systems and control modalities.

### 4.1. User Study Design

For the final user study, eight study participants were chosen via convenience sampling, including other course members, friends and family. Participants first filled out a questionnaire about themselves. 75 % percent of participants were in the age group of 20 - 30 and 50 % of participants had previously used mixed reality interfaces.

The actual study was then performed, for which the subjects put on the HoloLens with the application running. They were then introduced to the application and asked to go through the tutorial themselves. After that they were asked to stack three cubes on top of each other within three minutes. The option of moving the working volume was disabled for this task. The cubes each had a side length of 7 cm and the gripper had a maximum opening width of 9 cm, giving a margin of error of 1 cm on both sides. Finally the subjects were asked to document their experience by answering the standardised questions of the short user experience questionnaire (UEQ) [5].

### 4.2. User Study Results

The data collected from the final user study is plotted in Fig. 9. From this, it is possible to conclude that participants rated the application highly in terms of inventiveness, and found it very interesting and exciting. They also mostly agreed that the application is efficient. While still mostly remaining on the positive side, the opinions of users diverged on how easy and supportive the system is. We found that users with previous experience with MR on average rated their experience as easier and less obstructive than those with no prior experience. We also asked students and instructors to fill out the UEQ survey during the demonstration after the final presentation. For this demonstration, the feature to move the working volume was enabled. One interesting observation we made, was that the median of the answers to obstructiveness and ease of use was lower by one point compared to the user study. Qualitative feedback that we received supported the assumption that the working volume was often moved on accident and did not behave as the users would have expected, further underlining that this aspect of the interaction needs to be improved. Concerning the effectiveness of the system, we found that four of the eight participants (50 %) were able to stack the three cubes. Two subjects were only able to stack two cubes (25 %) before the three minutes were over. Two subjects (25 %) did not manage to stack any cubes. During the demonstration, with no time constraint, about 90 % of the people that tried it were able to stack at least two cubes. Unfortunately, one factor that was not taken into account was the strength of the gripper, which in combination with the weight of the cubes was sometimes insufficient to hold on to the cubes. The statistical validity and accuracy of these results is debatable, as the sample size was small and potentially not representative. Another possible factor that might have skewed results, is that participants were often related in some way to the observers and may have given better scores out of kindness.

### 4.3. Accuracy

No direct experiments were conducted for assessing the accuracy of the controller. Positional and rotational errors are introduced by inaccuracies of the hand pose estimation and errors in the reported robot joint angles. Both of these errors are however too small to be perceived with the naked eye and small compared to the error introduced by the *deadzone* added to remove jitter, as laid out in Sec. 3.3. This *deadzone* is chosen to allow for a maximum positional error of 1.4cm in each dimension and a rotational error of 1.2 degrees around each axis. In practise, we observe the error to be less than these tolerances most of the time, as the robot typically still has some momentum when entering the *deadzone*.

## 5. Discussion

### 5.1. Limitations

The scope of the project was chosen to focus in particular on the intuitive interaction with the robot. Hence, one limitation was consciously introduced by the use of ArUco markers for the scene reconstruction. Thus, any object that is not marked cannot be identified and interacted with. Moreover, ArUco markers can only be used on parallelepipeds, further limiting what objects can be tracked. On top of that, ArUco marker detection is sensitive to lighting conditions of the environment. As only marked objects are being tracked, obstacles are not visualised to the user. Hence, collisions need to be avoided by removing untracked objects entirely from the interaction space.

According to its datasheet, the RealSense depth camera used to detect the ArUco markers is only able to obtain good depth estimates for a distance over ∼28 cm. Experiments show, that to detect the markers confidently at this distance, a marker size of at least 7 cm is needed. This in turn puts a lower bound on the size of objects that can be interacted with and led to a redesign of the robot gripper.

The qualitative feedback gathered during the user studies allows for the identification of key limitations regarding the user interface. As mentioned in Sec. 4.2, the gesture that moves the working volume was often accidentally triggered, which caused undesired behaviour of the robot. In general, users were surprised by sudden movements of the robot. This can potentially be solved by only beginning to follow the hand once it is in close proximity to the end effector. Additionally, users requested the option to give inputs via voice commands, which are a particularly good fit for entering and leaving the hand follow mode. As mentioned before, the robot currently only follows inputs from the right hand, a limitation which can be removed in the future by allowing users to define their dominant hand.

Finally, the application was designed around a stationary setup of the server, robot and camera. Obtaining the correct static transform between the robot and the camera is still a tedious process and requires manual tuning of parameters. Once the setup is completed, the camera needs to remain static with respect to the robot. Minimal changes can already cause a misalignment of the virtual and physical objects in the scene. Additionally, the ROS server should ideally be assigned a static IP address, as changing it requires to rebuild the application for the HoloLens with the updated connection settings.

### 5.2. Future Work

Future points of work should focus on the main limitations of the application. In order to enhance the user experience, future work should continue to focus on improving intuitiveness and predictability of the system. Giving inputs via voice commands would be an important addition, especially since this would free up the user's dominant hand entirely for the task of controlling the robot. Voice commands could also potentially address the unwanted behaviour of the working volume.

Furthermore, depending on the application, the safety of the controller could be further improved by adding collision detection and possibly obstacle avoidance.

To speed up the initial setup and remove the requirement for the system to remain stationary, one could implement an automatic and ongoing estimation of the transformation between robot and camera.

Future work could also combine the custom controller developed in this work with a VR environment and a 3D camera mounted on top of the robot arm, in order to interact with an arbitrary 3D scene. This would remove the limitation of only being able to interact with predefined objects, but might introduce delay and cognitive saturation. On the other hand, if the objects to manipulate are known, the marker based tracking could be replaced by promising machine learning-based object detection [8].

## 6. Conclusions

We have presented an intuitive MR application for the teleoperation of a robot arm utilising a standard RGB-D camera, a HoloLens 2 and an off-the-shelf robot manipulator. Our proposed control interface enables almost any non-trained operator to remotely control a robot arm and solve a predefined assembly task solely by acting on a digital twin of the physical scene.

The conducted summative users study underlines the merit of the novel control in regards to the perceived efficiency and the ease of use of the application, even for users previously unfamiliar with MR devices.

There remain many limitations of our work. The marker-based detection algorithms used are specific to our devised problem setup and do not generalise well. We cannot deal with obstacles in our operational space at the moment and intuitively shifting the work space remains an open challenge. We also did not fully leverage the capabilities of the HoloLens 2 regarding the implementation of voice commands to extend the user input.

Nevertheless, this project lays the ground work for future instinctive robot control applications and novel human-machine interfaces based on virtual and mixed reality devices. It also serves to demonstrates the potential of these exciting and potentially fruitful research avenues for the research community at large.

## References


[1] Extend robotics, vr controlled robotic arm. https://www.extendrobotics.com. Last accessed: 2022-01-04. 2

[2] MoveIt. https://moveit.ros.org. Last accessed: 2022-01-07. 3

[3] Realsense aruco markers tracking, github repository. https://github.com/zptang1210/RealsenseArUcoTracking. 6

[4] Unity robotics hub, github repository. https://github.com/Unity-Technologies/Unity-Robotics-Hub. Last accessed: 2022-01-04. 2, 6

[5] User experience questionaire. https://www.ueq-online.org. Last accessed: 2022-01-05. 7

[6] Alex Adkins, Lorraine Lin, Aline Normoyle, Ryan Canales, Yuting Ye, and Sophie Jörg. Evaluating grasping visualizations and control modes in a vr game. *ACM Transactions on Applied Perception*, 18(4):1–14, 2021. 3

[7] Sebastian Blankemeyer, Rolf Wiemann, Lukas Posniak, Christoph Pregizer, and Annika Raatz. Intuitive robot programming using augmented reality. *Procedia CIRP*, 76:155–160, 2018. 1

[8] Zhaowei Cai, Quanfu Fan, Rogerio S. Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, page 354–370, 2016. 8

[9] Sonia Mary Chacko and Vikram Kapila. An augmented reality interface for human-robot interaction in unconstrained environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3222–3228. IEEE, 2019. 1

[10] Mark Draelos, Brenton Keller, Cynthia Toth, Anthony Kuo, Kris Hauser, and Joseph Izatt. Teleoperating robots from arbitrary viewpoints in surgical contexts. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. 1

[11] John L. Crassidis F. Landis Markley, Yang Cheng and Yaakov Oshman. Averaging quaternions. *Journal of Guidance, Control and Dynamics, Vol. 30, No. 4*, July-August 2007. 6

[12] Samir Yitzhak Gadre, Eric Rosen, Gary Chien, Elizabeth Phillips, Stefanie Tellex, and George Konidaris. End-user robot programming using mixed reality. In *2019 International conference on robotics and automation (ICRA)*, pages 2707–2713. IEEE, 2019. 1

[13] Andre Gaschler, Maximilian Springer, Markus Rickert, and Alois Knoll. Intuitive robot tasks with augmented reality and virtual obstacles. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6026–6031, 2014. 2, 3

[14] Jan Guhl, Son Tung, and Jörg Kruger. Concept and architecture for programming industrial robots using augmented reality with mobile devices like microsoft hololens. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4, 2017. 2

[15] Kangsoo Kim, Mark Billinghurst, Gerd Bruder, Henry Been-Lirn Duh, and Gregory F Welch. Revisiting trends in augmented reality research: A review of the 2nd decade of ismar (2008–2017). *IEEE transactions on visualization and computer graphics*, 24(11):2947–2962, 2018. 1

[16] R. Marin, P.J. Sanz, P. Nebot, and R. Wirz. A multimodal interface to control a robot arm via the web: a case study on remote programming. *IEEE Transactions on Industrial Electronics*, 52(6):1506–1520, 2005. 2

[17] Jakob Nielsen. *Usability Engineering*. San Diego: Academic Press, 1994. 2

[18] T. Pettersen, J. Pretlove, C. Skourup, T. Engedal, and T. Lokstad. Augmented reality for programming industrial robots. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pages 319–320, 2003. 2

[19] Jan Rosell, Raúl Suárez, and Alexander Pérez. Safe teleoperation of a dual hand-arm robotic system. *ROBOT2013: First Iberian Robotics Conference Advances in Intelligent Systems and Computing*, page 615–630, 2014. 2, 3

[20] Matthias Schröder, Christof Elbrechter, Jonathan Maycock, Robert Haschke, Mario Botsch, and Helge Ritter. Real-time hand tracking with a color glove for the actuation of anthropomorphic robot hands. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 262–269, 2012. 2, 3

[21] Yun-Peng Su, Xiao-Qi Chen, Tony Zhou, Christopher Pretty, and Geoffrey Chase. Mixed reality-enhanced intuitive teleoperation with hybrid virtual fixtures for intelligent robotic welding. *Applied Sciences*, 11(23):11280, 2021. 1

[22] 3D Systems. 3d systems, haptic input device "touch". https://www.3dsystems.com/haptics-devices/touch. Last accessed: 2022-01-04. 1, 3