Virtual Reality Framework for Better Human-Robot Collaboration and Mutual Understanding

Maciej K Wozniak* and Patric Jensfelt

Abstract— **Humans interact with robotic systems on a daily basis. User-friendly and efficient interfaces connecting us with these systems are critical for efficient collaboration and a good user experience. In the latest machine learning developments, many robotic platforms have used deep learning models to understand the environment and surroundings better. However, what a robot senses and how it takes decisions are usually hidden from the user. It is believed that soon we will be able to work side-by-side with these machines in a connected, collaborative space. Thus, it is essential to understand the robot and easily reason with it about the state of the environment or how it wants to execute a particular task. This work presents a virtual reality (VR) framework for human-robot collaboration, focused on improving communication and understanding between humans and robots.**

I. Introduction

Understanding the environment is one of the crucial tasks most robots must perform. The goal is not to hard-code each possible action, but to allow the robot to reason about the surroundings and learn how to move and act within a changing environment. Even though the goal may vary from delivering a package [1] to cleaning an apartment [2], comprehensive knowledge of the environment is required to complete the task. Another important factor is mutual trust and understanding between the user and the robot, so both can reason and collaborate safely and efficiently. Therefore, it is critical to see how the robot perceives the environment.

An interaction between a user and a robot may take various forms. An example of a commonly used one is verbal communication [3]. A user may ask a robot to move an object (e.g., a tennis ball) in a specific space (e.g., a room). To accomplish this (i.e., move the ball from point A to B), the robot has to understand the sentence (e.g., using natural language processing), execute the action and update its understanding of the environment. However, many errors may arise from an inaccurate sentence, e.g., *remove the ball from the table*; the robot may not understand *which ball should be moved* or *where it should be placed*. Such confusion may cause the whole system to fail [3].

To overcome these challenges, we propose an intuitive VR interface where a user can easily communicate with the robot. We create a 3D virtual representation of the real world, which the user sees and interacts with in VR. An interactive virtual interface shows what the robot understood about the environment in real-time so that the user can correct the robot's reasoning and actions if necessary (described in detail in Section V). This approach creates a clear understanding of the task and helps to eliminate possible errors. Now, the user can grab an object in the virtual environment and place it somewhere else as a way to instruct the robot about what to do in the real world. Because the virtual environment was created based on the output of the robot's perception module, the user understands what the robot knows about its workspace and can seamlessly communicate to the robot what actions it has to perform.

Additionally, the trajectory and movement of a robot may not always fit the users' preferences. For example, a robot is usually biased to choose the shortest path; however, humans' preferences of the robot's actions may differ. They can feel afraid and uncomfortable being side-by-side with the robot, not knowing what it will do if they are not used to being around a robot. Consequently, they may prefer to see what the robot intends to do beforehand so that they can approve or disapprove its actions and potentially modify them. If users could do that, they would most likely be less hesitant to work and share a common space with it.

Building trust and creating seamless human-robot interaction was the primary motivation for our work. Seeing precisely what a robot understands and what actions it will take brings us to another level of trust, which often cannot be achieved even in human-human interaction.

Our main contribution is a virtual reality framework for human-robot collaboration. Our tool serves the purpose of building users' trust through understanding how the robot perceives its surroundings and reasoning with it about its actions. Its main features are the ability to visualize what the robot understands about the environment, interact with it and alter its actions through an immersive VR user interface (UI).

We tested the framework within a virtual environment using a simulation engine and VR headset. We emphasize that an immersive VR UI elevates the interaction and communication between the user and the robot. VR makes it easy and intuitive to understand and perform spatial tasks such as 3D trajectory modification. Our work promotes the use of VR for human-robot collaboration and shows yet another application of VR in robotics-oriented projects.

II. Related Work

In this Section, we will review some of the recent projects focused on improving human-robot collaboration using virtual/augmented reality (VR/AR) devices. Reviewed articles also show how important it is for users to be able to reason,

*This research was supported by Wallenberg AI, Autonomous Systems and Software Program (WASP). Contact info: {maciejw,patric}@kth.se, KTH Royal Institute of Technology, Stockholm 10044, Sweden

understand, and discuss with the robot about its actions and intentions.

In recent years, researchers realized that VR elevates the interaction and communication between the user and the robot, making it an immersive experience. Szafir and Szafir [4] focused on data visualization between humans and robots, showing that data visualization is a fundamental aspect of good collaboration and mutual understanding between humans and robots. Other approaches focused on creating VR/AR frameworks for Human-Robot Collaboration (HRC). Mara et al. created Cobot [5] – an educational HRC platform where participants can play interactive mini-games with robots and work together on different tasks. Kennel-Maushart et al. [6] developed a tool that enables the user to manipulate robotic arms in the *real world* by applying force to them in the virtual reality setup. It is vital that when humans and robots work in the same space, robots can correctly estimate humans' locations and poses. VR headsets and controllers can greatly facilitate that task. The robot can easily track human movements using the sensors in the controllers and the headset. At the same time, the environment can be augmented and presented to the user in a different form via the headset, making the interaction more immersive and interesting [7].

These projects promote the use of VR for human-robot interaction and educate humans about robots; however, they do not focus on executing practical tasks or improving how humans interact with robots. On the other hand, our framework focuses on improving the interactions and understanding a robot's intentions for practical tasks.

Other researchers applied VR to projects in the industrial setting, creating and training specific scenarios in the virtual environment before performing them in the real world. An example could be the assembly process [8], collaborative tasks performed together with the robot [9], or visualizing safe space for physical assembly workers and robots [10].

These methods focus primarily on personnel training for manufacturing industry. Additionally, they are made in a fixed environment, whereas our framework can operate in changing environments and be used for various tasks.

Moreover, many scientists used VR to create and conduct *digital twin* experiments. A *digital twin* is a virtual representation of a robot. Modern physics engines can imitate reality in great detail, allowing a digital twin to be an accurate *test-bed* for real-world applications [11]. Many scientists, such as Kuts et al. [12] used VR in the development of a framework aimed to bridge the gap between real-world and simulation-based industrial robots. Others also showed various applications and benefits of using VR interfaces for digital twin projects, such as improved factory safety or workers' training [13].

Described projects focus on visualization and representation of a robot's action; however, they do not provide an easy-to-use interface for reasoning and correcting the robots which operate in a fixed environment. On the other hand, our work is focused on immersive and easy-to-use interface in which we can modify robot's actions, Additionally, our perception module allows a robot to function in a changing environment.

Finally, VR/AR has started to be commonly used in robot control and teleoperation tasks [14]–[16]. Articles by Ostatin [17] and Togias [18] showed that VR is perfect for planning the trajectory of the robotic arm, allowing the users to easily plan how the robot should move. Chandan et al. [19] developed a teleoperation framework that can be used to visualize the states, intentions, and future trajectories of robots. Xu et al. [20] efficiently visualize the state of the robot's end effector in VR, allowing the user to give orders by changing the end-effector position and orientation in the VR UI.

These projects focus on planning robots' trajectories, visualizing robots' intentions, and steering the robot; however, they do not focus on the explainability aspect of what robots understand about the environment. Their visualization and manipulation are limited to either only choosing the final position of the robot's end-effector or adding *no-go* zones. Additionally, none of these projects allow the user to test their solution before deploying them to the real world. On the other hand, our framework allows the user to see how the robot perceives the environment, modify every step of its action, and verify proposed actions and trajectories before deploying them to the real world.

III. Problem formulation

The projects described in Section II are a significant contribution to the research community. However, we still see missing parts that are addressed in our method. A framework for human-robot collaboration should emphasize the importance of understanding a robot's intentions and its perception of the working space, something we did not explicitly see in other solutions.

A framework interface should help reduce potential errors and misunderstandings, as well as increase the level of trust between users and robots, which is essential for long-lasting collaboration. Users should be able to interact with robots without explicit programming knowledge. An interface should be immersive, simple, and intuitive to use. It should allow users to quickly test different scenarios, visualize and modify the robot's intentions (e.g., the trajectory), and effortlessly deploy the final solutions to the real world.

To sum up, a fully functional solution should enable users to:

- help the robot to avoid collision with different obstacles by modifying its path.
- incorporate the preferences about the movement of the robot.
- see what robot does and what it does not understand about the environment.
- assist it with difficult and demanding tasks where it is more likely to fail.
- give tasks to the robot.

IV. PROPOSED FRAMEWORK ARCHITECTURE

In this Section, we describe the system architecture (shown in Fig. 1). Our approach takes an important step toward a final solution for the HRC framework, providing the user with a transparent representation of how the robot perceives its surroundings and the ability to visualize and modify its intentions. The user can test different scenarios and solutions in the virtual environment before deploying them to the real world. Our framework enables the user to interact with the robot throughout the immersive user interface, giving the robot clear commands, e.g., asking it to pick and move the objects by rearranging them in the VR. Finally, we provide the functionality for visualizing and correcting the robot's trajectory by simply grabbing and moving trajectory waypoints (intermediate points on the robot's path).

A. VR environment

We use the Unity game engine[1] to create a VR environment and test it using the Oculus Quest 2 headsets[2]. However, the framework can be built and run on another headset by changing the target device in Unity settings. In the rest of the text, we refer to the VR part of our framework as Virtual Reality User Interface (VR UI).

B. Environment mapping and understanding

In order to understand the environment, we used an RGB camera to collect data. The images are fed into a deep learning model (such as Detectron2 [21]) for scene segmentation and object detection to obtain segmentation masks and bounding boxes of the detected objects. This information will be later used to create and update the state of the environment.

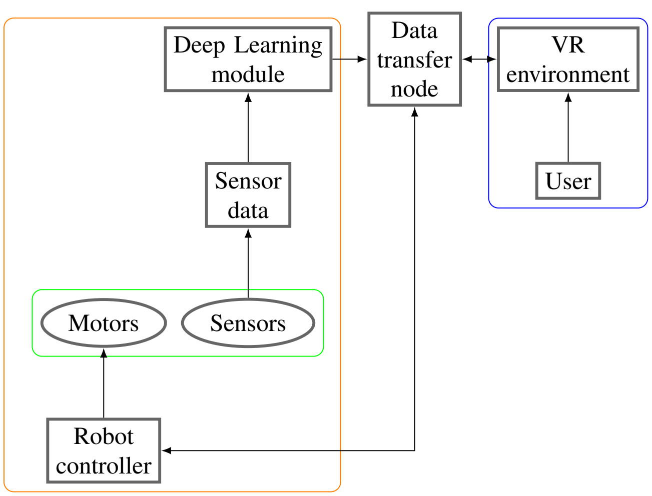


Fig. 1: Proposed system architecture. The elements within the orange box correspond to nodes and parts connected with the robot's motion and perception, whereas the ones in the green box are explicitly corresponding to its hardware. Nodes inside the blue box are connected to the user and VR setting. The robot's sensors collect data from the environment.

[1] https://unity.com/

[2] https://store.facebook.com/se/quest/

C. Data transfer

The first naive solution would be to transfer the data collected by the robot directly to the VR UI and show it to the user. However, such an approach creates various bottlenecks. It is computationally expensive to receive and render a high-quality environment in real-time. Moreover, if we use RGBD sensors or LiDAR instead of a camera, we would obtain point cloud data which is often incomplete. Thus, even though a robot understands from an image or a point cloud that an object on the table is a cup, it collects only the points from one side of the cup. Therefore, the model shown to the user in the VR would be incomplete and have multiple imperfections, such as numerous missing points. One can argue that we can solve it with shape completion or overlapping masks of the 3D models onto classified objects with those missing points. However, the lower quality of the interface could potentially worsen the overall user experience, making them more hesitant to use the tool. Additionally, such a solution would require much higher data transfer capacity or computational resources.

In order to facilitate the exchange of information, improve the user experience, and minimize the necessary bandwidth, we only transfer output of a detection network between the robot and the virtual environment. When the robot detects and classifies an object, it sends its class, location, and estimated size to the VR UI application. In the virtual environment, we have multiple prefabs (3D models of the objects we built into the project) corresponding to the detected classes, and we can quickly create a 3D representation of the room from the received message. Similarly, by reorganizing or pointing at the objects in the VR environment, we can send the request to the robot to change the position of that object in the real world. Seeing these objects and interacting with them in the VR UI help us to grasp a better understanding of how the robot perceives its environment.

V. EXPERIMENTS

The current experimental setup is shown in Fig. 2. We based our experiments on Niryo One robotic arm[3]; however, any robotic arm can be loaded into our framework using URDF and mesh files[4]. The environment is set up as follows. A robotic arm is located on the table with various objects placed on top. The camera is located above the table, and its output is sent to the ROS [4] node that runs the instance detection and classification algorithm. The information obtained by the DL model is then sent to the user interface on the VR headset and the objects are created in the exact location as they were detected and classified.

Our work focuses on the VR UI part of the system, therefore, to facilitate the experiments, we created a Unity simulation corresponding to the *real-world environment*. This approach enables us to easily generate the data necessary to train the deep learning model and run the experiments in a highly controlled environment. For the machine learning

[3] All the prefabs are available in the Unity Robotics Hub

[4] http://wiki.ros.org/Documentation

part, responsible for object detection and scene segmentation, we chose *Detectron2* with Faster R-CNN [22] pre-trained on COCO dataset [23]. We retrained it on the data generated in Unity using open-source prefabs (3D models such as water bottles or flashlights)[3]. The proposed framework and datasets are open-source and can be freely downloaded and tested[5].

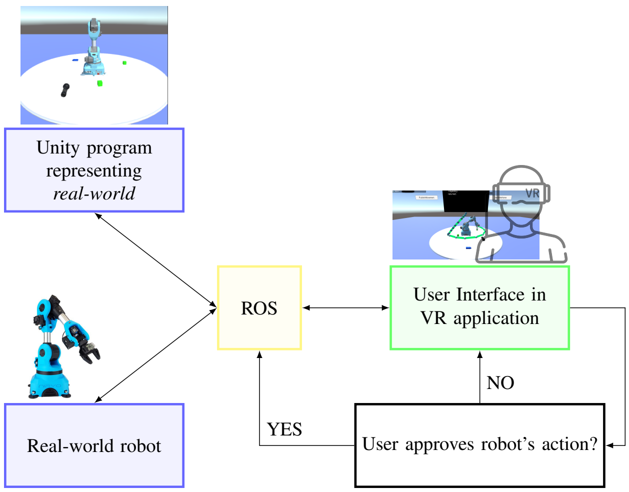


Fig. 2: Representation of experimental setup.

A. Environment understanding

In the first step, the virtual environment is created and the objects, detected and classified using the camera output from the real world, are generated in the VR UI. The user understands how the robot perceives its workspace because the interface reflects the output of the robot's perception module. If the robot's operator can only see the camera output (as in Fig. 3b) without feedback from the robot's perception module, the whole interaction may fail. A straightforward example could be a user asking a robot to pick up a tennis ball from the table. The robot may not have detected the object (e.g., a tennis ball) that the operator asks it to pick up (e.g., because the network was not trained on tennis balls Fig. 3a). Consequently, the robot will not understand that there is an object to pick up. That may lead to confusion and frustration from the users' side since they will not understand why the robot fails. Our method allows the user to see how the robot perceives the world and quickly identify such issues. The user can only interact with the objects that were correctly detected by the robot, as shown in Fig. 3c.

B. Action verification and safety

In the VR UI, objects can be selected by pointing at them with the controller as a laser pointer. Instead of testing the solution beforehand, one could quickly send such a command to a real-world robot. However, there may be potential flaws in the robot's trajectory. A simple scenario is an obstacle on the robot's course as a bottle in Fig. 4a. If we were to deploy the robot's action to the real world, we could potentially damage the robot. Instead, our framework enables users to verify and approve the robot's trajectory before deploying it into the real world. The proposed trajectory is generated and shown in VR. Additionally, a waypoint is generated for every timestep t (value defined by the user). The user can modify the frequency and appearance of the waypoints. We use the robot's end-effector Fig. 4b or a sphere Fig. 4c to show the waypoints, but any other 3D model can be assigned to do so. Now, the user can move around and quickly see the 3D trajectory from a different perspective, which would be more complex to do using a 2D screen.

C. Trajectory modification

When we send a request to the robot, it executes the task in the VR or reports that it is not able to make a move (e.g., the target is beyond its reach). As shown in Fig. 5a, the users can see that the initial trajectory would cause a collision with another object. In that case, the proposed trajectory can be disapproved and modified. In the VR UI, the user can move the waypoints so that the robot would take a different path and avoid collision with the obstacle, as shown in Fig. 5b. The user verifies a new trajectory and the robot can execute the task following a new path (Fig. 5c). Users can examine whether now the robot executed the moves as expected and, if necessary, repeat the correction process multiple times. As we can observe in Fig. 5d, the robot does not exactly follow the new trajectory. To provide fluid motion, we minimize the number of waypoints the robot has to pass. We only include the starting and final points and the waypoints which positions were changed. That allows us to achieve a smoother transition between poses while still avoiding the obstacles[6] Finally, when the trajectory is approved, it can be deployed and executed by the robot in the real world.

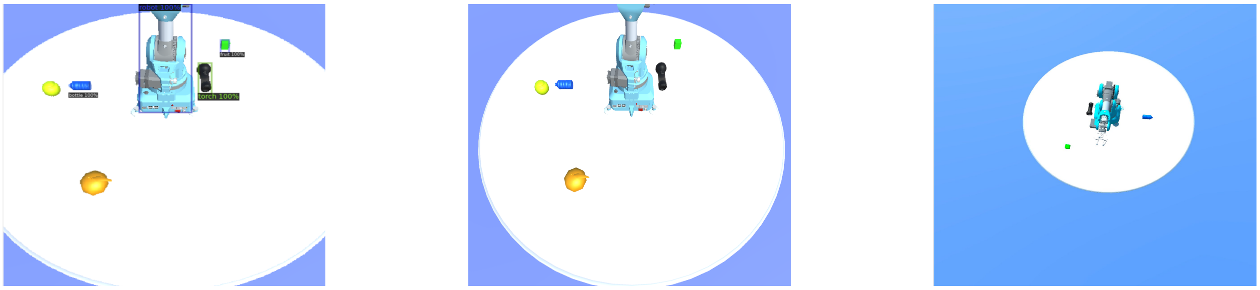
These tests allow us to eliminate potential malfunctions or hazardous situations. Additionally, we can visualize what the robot understands about the environment and its intentions in real-time. An option to review the robot's actions is essential, especially with the growing demand for home robotics. Robots are not anymore expected to work alone but to share space and collaborate with humans. Our tool will help humans feel safe and comfortable while working side-by-side with the robot.

VI. Conclusions and Future Work

In this paper, we presented a VR-based HRC framework and showed its capabilities in a number of tasks. In Section I, Section II, and Section III, we stated the problem of clear understanding between the user and the robot and presented existing solutions similar to ours, however, most of them lacked explainability of robot's intentions and environment understanding as well as the ability to modify actions proposed by the robot. Inspired by that, we created a HRC framework described in Section IV. In Section V, we showed how we created a representation of the real world in VR UI. Next, we presented how users can easily visualize and

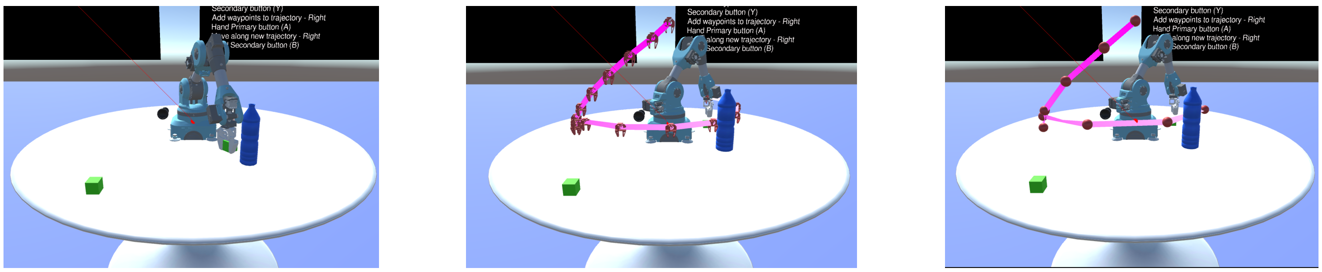
[5]Our framework, training data and DL model are publicly available for download and testing `https://github.com/maxiuw/pickandplace`

[6]A short `https://youtu.be/hs3DXQhG8Ys`. video showing an entire interaction with a robot.



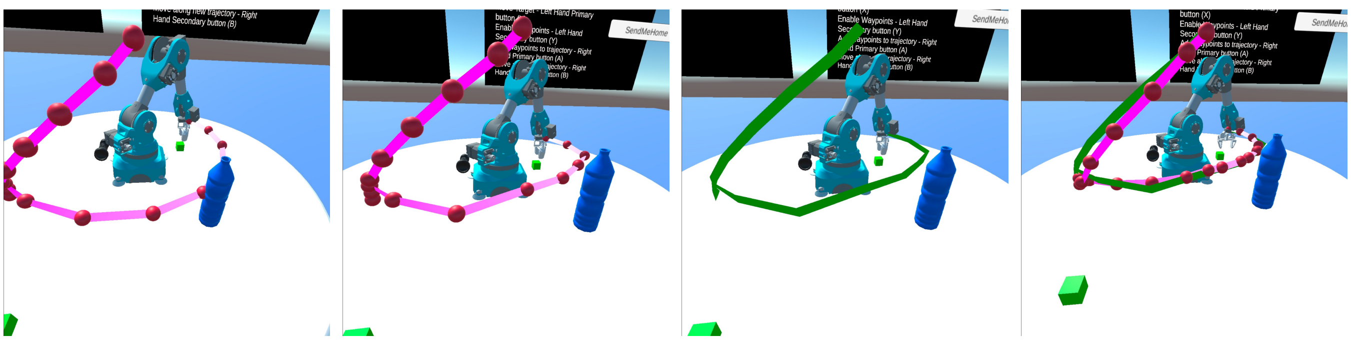
(a) Output of perception module (b) Camera view (c) View of the user in VR

Fig. 3: Potential issues caused by robot's perception flaws.



(a) Failure due to collision. (b) Trajectory view with end-effectors as a waypoints. (c) Trajectory view with spheres as a way-points.

Fig. 4: Collision caused by the the obstacle on the robot trajectory. We can see that if we deploy task into the real world, without verifying it before hand, the robot may fail to complete the task (Figure a).



(a) Proposed trajectory with collision. (b) Trajectory rearranged so that the robot avoids the obstacle. (c) Visualization of the new trajectory without waypoints. (d) New trajectory executed by the robot avoiding the bottle.

Fig. 5: Example of rearranging robots trajectory.

modify the robot's actions (e.g., to avoid obstacles) and deploy it back to the real world. Our immersive VR UI confirmed that VR is an excellent tool for interacting and collaborating with the robot. We emphasized the importance of understanding the robot's actions and abilities. While we run the experiments on a *pick–and–place example*, interacting with the robot can teach the user which tasks the robot has problems with. These aspects are fundamental, especially nowadays, when mobile robots are slowly moving to our offices [24] or homes [25].

There are still remaining challenges that we would like to tackle. First, it would be interesting to test our solution in AR setup instead of VR. This approach would allow users to see what is around them in the real world while interacting with the robot without the need to move to a fully immersive environment. Since the project was developed in Unity, it is simple to change the target device for the software; therefore, building it on an AR headset should not be an issue.

Additionally, we would like to conduct a user study on a larger group of people to get feedback on the methods we developed and adjust our approach for the interaction to be intuitive and user-friendly. Such a study will aim to show that knowing what actions the robot will take and how it perceives the environment helps the user to gain trust in the robot's abilities.

To sum up, this project presents the framework's ability for HRC and highlights the importance of common human-robot understanding.

REFERENCES

[1] M. Thiel, M. Grote, M. Schrick, and S. Tjaden, "Integrierte mobilität und automatisierung-transportroboter unterwegs im tabula shuttle/integrated mobility and automation-transport robots on the road in the tabula shuttle," *Automobiltechnische Zeitschrift*, vol. 124, no. 4, 2022.

[2] K. Doelling, J. Shin, and D. O. Popa, "Service robotics for the home: A state of the art review," in *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*, 2014, pp. 1–8.

[3] J. Peltason, N. Riether, B. Wrede, and I. Lütkebohle, "Talking with robots about objects: A system-level evaluation in hri," in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2012, pp. 479–486.

[4] D. Szafir and D. A. Szafir, "Connecting human-robot interaction and data visualization," in *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 2021, pp. 281–292.

[5] M. Mara, K. Meyer, M. Heiml, *et al.*, "Cobot studio vr: A virtual reality game environment for transdisciplinary research on interpretability and trust in human-robot collaboration," 2021.

[6] F. Kennel-Maushart, R. Poranne, and S. Coros, "Multi-arm payload manipulation via mixed reality,"

[7] B. Yang, X. Xie, G. Habibi, and J. R. Smith, "Competitive physical human-robot game play," in *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 2021, pp. 242–246.

[8] J. Guzzi, G. Abbate, A. Paolillo, and A. Giusti, "Interacting with a conveyor belt in virtual reality using pointing gestures," in *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '22, Sapporo, Hokkaido, Japan: IEEE Press, 2022, pp. 1194–1195.

[9] V. Ortenzi, M. Filipovica, D. Abdlkarim, *et al.*, "Robot, pass me the tool: Handle visibility facilitates task-oriented handovers," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2022, pp. 1–9.

[10] J. Zubizarreta, I. Aguinaga, and A. Amundarain, "A framework for augmented reality guidance in industry," *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 9, pp. 4095–4108, 2019.

[11] A. A. Malik and A. Brem, "Digital twins for collaborative robots: A case study in human-robot interaction," *Robotics and Computer-Integrated Manufacturing*, vol. 68, p. 102 092, 2021.

[12] V. Kuts, T. Otto, T. Tähemaa, and Y. Bondarenko, "Digital twin based synchronised control and simulation of the industrial robotic cell using virtual reality," *Journal of Machine Engineering*, vol. 19, 2019.

[13] X. Wang, C. J. Liang, C. C. Menassa, and V. R. Kamat, "Interactive and immersive process-level digital twin for collaborative human–robot construction work," *Journal of Computing in Civil Engineering*, vol. 35, no. 6, p. 04 021 023, 2021.

[14] F. Nenna and L. Gamberini, "The influence of gaming experience, gender and other individual factors on robot teleoperations in vr," in *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, 2022, pp. 945–949.

[15] A. Naceri, D. Mazzanti, J. Bimbo, *et al.*, "Towards a virtual reality interface for remote robotic teleoperation," in *2019 19th International Conference on Advanced Robotics (ICAR)*, 2019, pp. 284–289.

[16] E. Rosen, D. Whitney, E. Phillips, D. Ullman, and S. Tellex, "Testing robot teleoperation using a virtual reality interface with ros reality," in *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, 2018, pp. 1–4.

[17] M. Ostanin and A. Klimchik, "Interactive robot programing using mixed reality," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 50–55, 2018.

[18] T. Togias, C. Gkournelos, P. Angelakis, G. Michalos, and S. Makris, "Virtual reality environment for industrial robot control and path design," *Procedia CIRP*, vol. 100, pp. 133–138, 2021.

[19] K. Chandan, V. Kudalkar, X. Li, and S. Zhang, "Arroch: Augmented reality for robots collaborating with a human," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 3787–3793.

[20] S. Xu, S. Moore, and A. Cosgun, "Shared-control robotic manipulation in virtual reality," *arXiv preprint arXiv:2205.10564*, 2022.

[21] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, *Detectron2*, `https://github.com/facebookresearch/detectron2`, 2019.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[23] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[24] M. Hanheide, M. Göbelbecker, G. S. Horn, *et al.*, "Robot task planning and explanation in open and uncertain worlds," *Artificial Intelligence*, vol. 247, pp. 119–150, 2017.

[25] G. A. Zachiotis, G. Andrikopoulos, R. Gornez, K. Nakamura, and G. Nikolakopoulos, "A survey on the application trends of home service robotics," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2018, pp. 1999–2006.