Assessing the Effectiveness of Augmented Reality Handheld Interfaces for Robot Path Programming

Francesco De Pace[1(✉)], Andrea Sanna[1], Federico Manuri[1], Damiano Oriti[1],
Simone Panicucci[2], and Valerio Belcamino[1]

[1] Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Turin, Italy
{francesco.depace,andrea.sanna,federico.manuri,
damiano.oriti,valerio.belcamino}@polito.it
[2] COMAU S.p.A, Via Rivalta, 30, 10095 Grugliasco, Italy
simone.panicucci@comau.com

Abstract. The Industry 4.0 is changing the very nature of the factories. If, in the past, the industrial facilities were composed by a plethora of devices disconnected from each other, nowadays, the digitalization of the industrial processes allows the devices to continuously exchange data with each other, thus improving the levels of production. Autonomous robots represent one of the main Industry 4.0 pillars. Among them, the collaborative robots are expected to be extensively used on the production lines allowing the companies to face an increasingly competitive market. In order to not be cut off from the production process, the human operators should combine their intrinsic capabilities of adapting to unforeseen scenarios with the high level of accuracy, speed and strength of the robots. Augmented Reality (AR) is another main pillar and it results to be effective for programming the robotic arms. Several AR interfaces have been evaluated in such scenarios in the last decade. However, there is a lack of studies that have deeply analyzed the effectiveness of AR handheld interfaces to control arm robot for tasks that require the creation of robot path in the 3D space. The study presented in this work fits in this context, proposing an evaluation of an AR handheld interface to control a robotic manipulator. The main results suggest that these types of interfaces are slightly adequate for controlling the manipulators, indicating that there is still room for improvements and research.

Keywords: Augmented Reality · Robot path programming · Collaborative robotics · Cobots

1 Introduction

So far, our society went through three different industrial revolutions. At the end of the 18th century, the society moved from an agriculture-based society

Z. Lv and H. Song (Eds.): INTETAIN 2021, LNICST 429, pp. 336–352, 2022.
https://doi.org/10.1007/978-3-030-99188-3_21

to a mechanical one kicking off to the first industrial revolution. The discovery of electricity about a century later ushered in the second industrial revolution, bringing with it revolutionary technologies such as the telegraph and the telephone. The third industrial revolution began in the second half of the twentieth century, marked by the advent of ICT technologies. If the three previous industrial revolutions were marked by mechanization, electrical energy and widespread digitalization [1], we are now on the verge of a fourth industrial revolution, named *Industry 4.0*, in which factories are supposed to be fully autonomous and intelligent. The German government coined the word *Industry 4.0* to describe a high-tech plan for potential manufacturing industries [2]. The idea of Smart Manufacturing is at the heart of this modern movement [3], implying that factories will become smart factories as new technologies such as the Internet of Things and Cyber Physical Systems become more widely adopted [4]. The nine main pillars of Industry 4.0 [4] reflect the nine main innovations that factories are encouraging and employing to enhance all aspects of the manufacturing process. One of the key innovations that can successfully help the fourth revolution is represented by Augmented Reality (AR). It is one of the major Industry 4.0 foundations and it has been used in a great variety of industrial tasks [5]. Another essential pillar is represented by the autonomous robots, which are supposed to boost and increase the efficiency of future factories. It is expected that the autonomous robots will start working side-by-side with the human operators [6], "becoming" the so called *collaborative robots* or *cobots* [7]. In order to reduce the risk of hazards, the industrial robots have traditionally worked in well-defined areas, completely separated by the human operators. The new AR interfaces, on the other hand, offer novel forms of interaction that can ensure the safety of human operators while also allowing to effectively control the robots through innovative interaction paradigms. So far, four different types of interfaces have been used to interact with the manipulators: (i) desktop, (ii) projected, (iii) handheld and, (iv) wearable. However, despite the related state of the art provides a fair amount of works and projects, there is a lack of studies that deeply investigate the effectiveness of the AR interfaces in the Human-Robot Collaborative (HRC) area from a user-centred perspective [7]. Since there will be a considerable increase in the number of robots employed on the production line [8], it is expected that the human operators will share the workspace with the robots and machines, thus collaborating with them and generating new forms of interaction and production [9]. Hence, it becomes of primary importance investigating and developing innovative interfaces aiming at improving the human-robot collaboration guaranteeing the safety oh the human workers. Moving from these considerations, this work presents an AR handheld interface to program robotic manipulators. The interface allows the users to create and manipulate robot end-effector paths in the real space. The interface has been deeply evaluated, collecting both objective and subjective data. The related results have been analyzed providing useful insights about the strengths and weaknesses of the handheld interfaces when used in the HRC area.

2 Background

AR technologies have been researched into three main HRC areas: (i) to visualize the robot workspace (*Workspace* area), (ii) to get a virtual feedback over the robot control (*Control Feedback* area) and (iii) to provide information regarding the task or the robot itself (*Informative* area) [7]. Since this work mainly focuses on the Control Feedback category, this section will presents only a detailed review of the state of the art strictly related to the aforementioned category. However, interested readers are encouraged to refer to [7] for a complete review of the main uses of the AR interfaces in the HRC domain.

Following the definition of the Control Feedback category [7], the AR interfaces are commonly used to (i) provide a feedback over the user's input itself and (ii) to visualize and manipulate the virtual robot paths. Regarding the former, a comparison between an AR gaze-based interface and a gesture based one is proposed in [10]. Both interfaces allow the users to select the real objects that will be manipulated by a robotic manipulator. The motion of a virtual representation of the robotic arm provides the users a feedback on the user's input. Similarly, in [11,12] the users can control a virtual robot using the Wiimote controller and a wearable AR device, respectively. Besides pointing out the motion of the robot through its virtual counterpart, also the real objects of interest can be highlighting using virtual metaphors. Frank et al. [13] proposed a handheld AR interface to control a robotic arm for a pick-and-place scenario. The device camera is used to capture a live video-stream and the users can select the object of interest by exploiting the touch capabilities of the considered device. Once selected, a virtual asset is superimposed on the real object, highlighting the user's input (the proposed work has been extended in [14] considering also egocentric and exocentric interfaces). The authors of [15–17] proposed a gesture-based AR interface to control a robotic arm for a pick-and-place task. Referring to their most recent work [17], the camera of a handheld device is used to recognize hand gestures that are translated into robot instructions. The AR system has been compared with a kinesthetic approach and with an offline programming method (refer to [18] for a comprehensive definition of the word *kinesthetic*). The main results show that the AR interface greatly lowered the task time and it was more appreciated by the users. The camera of a handheld device is also used in [19] to automatically recognize the real objects that should be manipulated by a robotic arm. The main results indicate that the proposed object detection algorithm is reliable enough to be effectively employed to program a real manipulator. Recently, Chacko and Kapila [20] introduced a smartphone AR interface to control a small-size manipulator. The system is similar to the one proposed in [13] and it has been deeply evaluated involving a considerable amount of users. The main outcomes indicate that the proposed interface allows the users to accurately control the robot arm. Furthermore, the interface obtained high usability scores and low workload scores, suggesting that the proposed solution could be effectively employed for pick-and-place scenarios. Additional works can be found in [21], which details a handheld AR interface to ease the creation of robot programs, and in [22] whose authors proposed an AR interface that allows users to create virtual assembly sequences that will be executed by a robot arm.

Considering the robot path, several works have employed desktop AR interfaces to manipulate virtual paths of robotic manipulators. It is worth noticing that in this work a *virtual path* is defined as a set of connected 3D points created by the user. Chong et al. [23] proposed an heuristic search solution to ease the creation and manipulation of virtual manipulator paths. The users can add, remove and modify the virtual path points using a flat image-based stylus tracked by an external camera. Once the path is defined, if a virtual point is not reachable by the real manipulator, the point color is changed to red to warn the user. Similarly, the accuracy of a desktop AR interface for path programming has been evaluated in [24,25]. The main results show that the interface can achieve an accuracy of 11 mm using a camera placed at 1.5 m from the workspace. Further projects that employ AR desktop interfaces to manipulate robot paths can be found in [26,27]. Projected and wearable interfaces have been also used to manipulate virtual robot paths. Zaeh and Vogle [28] proposed an interactive AR projected interface to program industrial robots. The users can define in the real space a set of virtual points and the related path is then executed by the real manipulator by using a tracked stylus. The outcomes indicate that the proposed system greatly reduced the task time with respect to a kinesthetic approach. Moreover, the adopted tracking methodology can achieve an accuracy of $\sim$0.5 mm. A similar approach is proposed in [29]. The projected AR interface allows the users to control a manipulator during grinding processes of ceramic parts. The main outcomes are in line with the work proposed in [28], showing a considerable reduction of the time required to program the robotic arm. Additional interesting works can be found in [30,31]. Regarding the wearable devices, the interface proposed in [32] provides the users the ability to manipulate the virtual path by exploiting the gestures recognition capabilities of the adopted device. Furthermore, the torques of each joint is also displayed in the real environment. Quintero et al. [33] compared an AR interface based on speech and gestures recognition with a traditional kinesthetic approach. Despite the proposed AR interface required less time to program the robot than the kinesthetic programming method, high levels of mental loads have been detected due to requirements of the speech interface to memorize a set of pre-defined vocal commands. A tracked stylus is used in combination with a wearable device to manipulate virtual paths in [34]. Similarly to the previous works, the proposed interface greatly reduced the execution time for welding and pick-and-place tasks (from 347 s to 63 s and from 117 s to 34 s, respectively). Further research projects regarding the use of wearable AR interfaces in the HRC domain can be found in [35,36].

3 Methodology

To the best of the authors' knowledge, the only work that analyzes the effectiveness of an AR handheld interface to manipulate virtual robot paths can be found

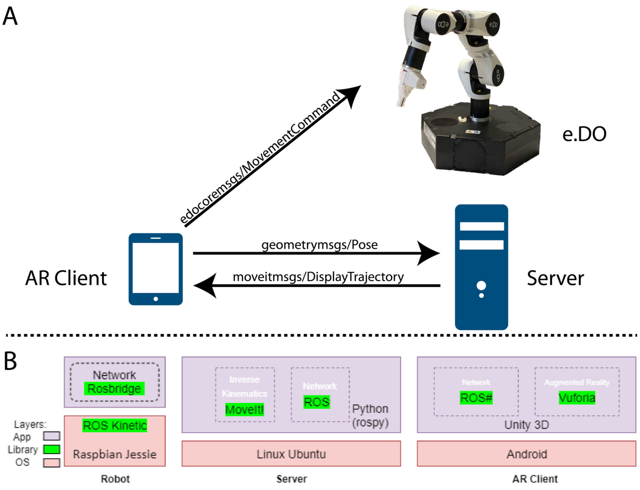


Fig. 1. A: the hardware architecture. B: the software architecture.

in [37]. The proposed solution allows the users to define virtual paths using a smartphone device. Although the interface has been assessed considering both objective and subjective parameters, it presents some limitations that should be overcome: (i) the creation of the virtual path is limited to 2D plane surfaces and (ii) the overall interface accuracy has been determined involving only one single user. The study presented in this work extends the analysis done in [37] by evaluating more challenging tasks, considering both 2D and 3D paths (e.g., virtual robot paths that are not constrained on 2D planar surfaces but they can be specified in the 3D space). Furthermore, the overall interface accuracy has been determined by considering several users, providing useful insights from a user-centred perspective analysis.

To achieve the mentioned goals, the following methodology has been adopted:

– development of an AR handheld interface that allows users to create virtual robot paths in the 3D space surrounding the robot;
– definition of a sequence of tasks to evaluate the proposed AR interface;
– definition of a proper metric to measure the accuracy of the AR interface;
– the choice of the appropriate questionnaires to measure the levels of usability and workload of the AR interface.

In the following section, the proposed system is detailed along with its hardware and software architectures. Furthermore, the handheld AR interface is presented discussing its main functionalities.

4 The Proposed System

The hardware and software architectures are introduced and detailed in this section. Moreover, the proposed AR interface is deeply discussed, describing its main functionalities.

4.1 The System Architecture

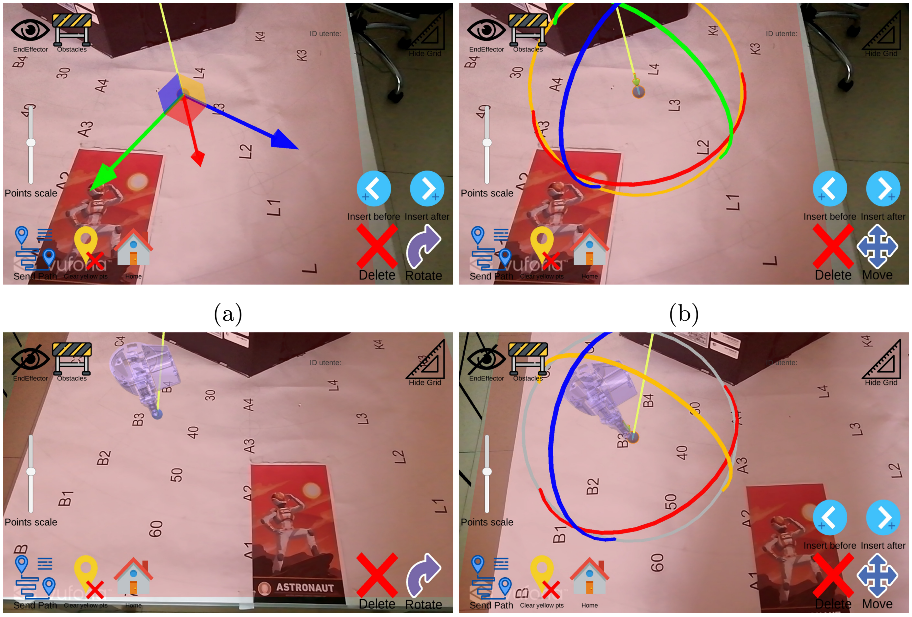
Figure 1 shows the system architecture. It is essentially composed by three distinct elements: (i) an Android tablet, (ii) a remote Personal Computer (PC) and (iii) a real manipulator. The tablet runs the AR interface developed using the Unity3D [38] game engine. The Vuforia SDK [39] has been integrated to provide a common reference system between the tablet and the manipulator. The alignment is carried out by tracking an image target positioned at a known location with respect to the real robot. The Unity3D application exchanges data with the remote PC that is responsible of computing the virtual robot path (see Sect. 4.2). The PC runs the Ubuntu 20.04.2.0 [40] distribution and the Robot Operating System (ROS) Melodic [41]. The real manipulator is represented by the COMAU e.DO [42] robotic arm. It consists of a 6 degrees-of-freedom (DOF) ROS-based manipulator equipped with a two-fingers robotic hand. The tablet, the PC and the manipulator are connected on the same Local Area Network (LAN) to effectively exchange data. The advantage of having the inverse kinematic (IK) solver decentralized (i.e., running at the server side) with respect to the real manipulator is straightforward: the propose architecture is *robot independent* and it can be easily extended considering different real manipulators.

The main functioning of the system is the following: when the user adds a new virtual point using the tablet device, the point position and orientation are sent through a TCP socket connection to the PC that in turns computes a possible path using an inverse kinematic algorithm provided by the MoveIt [43] ROS package. Once the path is computed, it is sent back to the Unity3D application and it is displayed in the real environment. Then, in case the user is pleased with the computed path, the application sends the acquired path to the real manipulator through an additional TCP socket connection, starting the motion of the real robot.

In the following section, a detailed description of the proposed AR interface is presented, illustrating the mechanism of virtual path creation and modification.

4.2 The AR Interface

The main functionalities of the proposed interface provide the ability of adding one ore multiple virtual points to the real scene and of visualizing two different versions of the virtual robot path. Specifically, when the user frames the image target, a virtual representation of the e.DO manipulator is superimposed on its real counterpart. However, it has been decided to keep visible only the robot end-effector to reduce the negative effects of possible occlusions. Both the robot



(a) (b)

(c) (d)

Fig. 2. (a) The VP translational tool. (b) The VP rotational tool. (c) The virtual robot end-effector. (d) The virtual end-effector during a rotational manipulation.

and the image target have been placed on a table with the base of the robot positioned at the same height of the target. The user can add a new virtual point (VP) to the real scene by touching the tablet surface at the desired location. In this work, a VP is represented by a virtual sphere carrying both the positional and rotational information required by the IK solver to compute the end-effector pose. In order to ease the VP addition, a virtual invisible plane is instantiated at the same height of the image target, preventing the user from adding a point below the table, that is, in a position not reachable by the real robot. The new point is added sequentially to the point list (a new point is inserted by default as last element of the list).

When a new VP is rendered into the scene, it is instantiated with a default orientation. Specifically, with the forward vector parallel to the surface normal direction (i.e., the end-effector perpendicular with respect to the surface). If the user selects one of the VPs, its three local axes are displayed and they can be used to drag the point into a new location (Fig. 2a). By changing to the orientation modality, a virtual gimbal is rendered and it can be used to change the VP orientation around its local axes (Fig. 2b).

In order to improve the visualization of the final pose of the robot end-effector, the interface can render a 3D model of the end-effector at the VP location (Fig. 2c). The transformations applied to the VP are mapped directly to the virtual end-effector (Fig. 2d).

To provide the user the ability of adding a VP between two other VPs, the user should select a point and use the arrow buttons positioned at the bottom right-side of the user interface. These two buttons allow the user to add a new VP at the *right-side* and *left-side* of the selected VP, respectively. Referring to Fig. 3, when the user wants to add a new VP at the left-side of P_1, it is verified that $D \geq s$, with D being the distance between two VP and s being the VP diameter, respectively. In case the condition is verified, a new VP of diameter s is rendered at $D/2$ (and the new VP is added to the point list at the specified position), otherwise the dimension of the VPs is iteratively reduced until a suitable position is found. The reduction it is constrained by $s \geq s_{min}$ to ensure a clear visualization of the VPs (the s_{min} value has been experimentally computed). In case the constrain is violated, a message appears informing the user that is not possible to add further VPs.

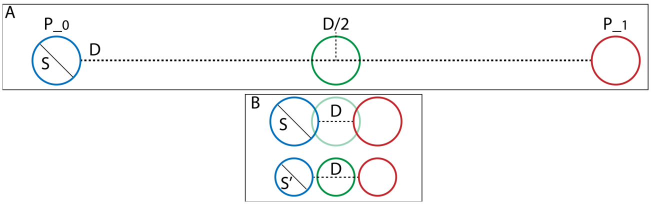
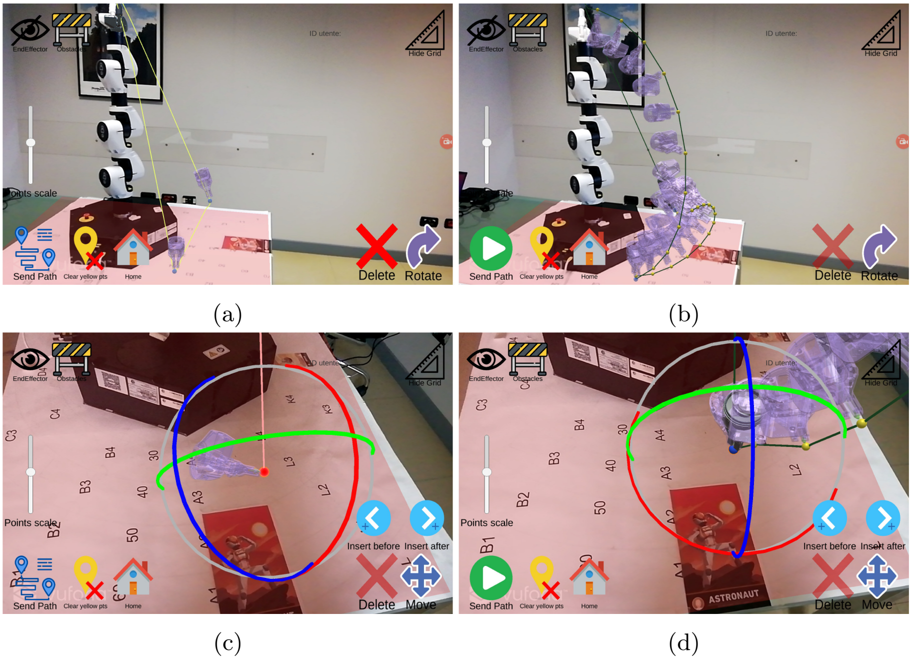


Fig. 3. The blue and red circles represent two VPs (P_0 and P_1, respectively) already added by the user. When the user wants to add a new VP (the green circle) at the left-side of P_1, there can be two different conditions: (a) when $D \geq S$, the VP is successfully added, (b) when $D < S$, the dimension of all the VPs is firstly scaled down and then the green VP is added between P_0 and P_1. (Color figure online)

The next step would be displaying the virtual robot path passing through the user's VPs. However, due to time complexity of the IK solver, the robot path cannot be computed in real-time (i.e., when a new point is added or modified). Hence, a Catmull-Rom virtual spline is firstly displayed starting from the current position of the robot end-effector and passing through the VP added by the user (this process is repeated every time a new VP is added to the scene). Although this spline only approximates the real robot path, it improves the user's understanding of the VP sequence.

Once all the VPs have been added to the real environment, the user can send them to the remote PC through the socket connection. Each VP describes the position and orientation that the end-effector should have in that specific point and the IK solver (running at the remote PC) tries to calculate a suitable path passing through all the VPs added by the user. As the path is computed, it is sent back to the AR application through the same socket connection and the approximated Catmull-Rom spline is modified according to the new path. Furthermore, the virtual robot end-effector starts moving along the virtual path, allowing the user to pre-visualize the movement of the real robot. Finally, the user can send to the real robot the new computed path, starting the real manipulator motion. Figure 4a and Figure 4b show the approximated and real robot virtual paths, respectively.



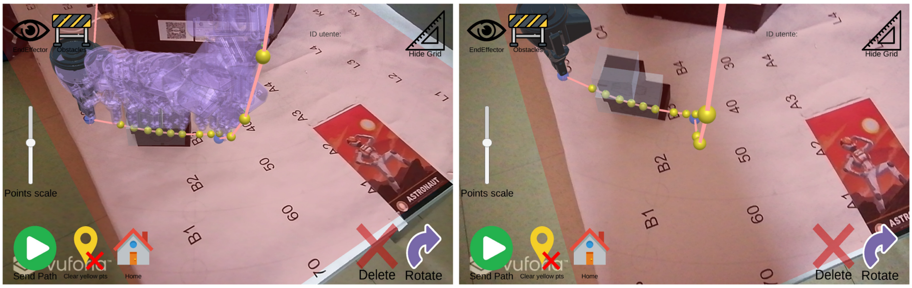
(a) (b)

(c) (d)

Fig. 4. (a) The approximated robot path is displayed by means of a yellow virtual line. (b) The same line is modified according to the real path and its color is changed to green. (c) The VP has been added with a wrong orientation. (d) The VP orientation has been successfully updated. (Color figure online)

If the user has added a VP in a position not reachable by the real manipulator, the IK solver would fail in computing a suitable path. In this case, the PC sends a message of error back to AR application that colors in red the VP that has been placed in a wrong position. A VP is considered in a wrong configuration when the end-effector cannot reach the VP position (in terms of X, Y and Z coordinates) or it cannot reach the VP position with the orientation (in terms of quaternion) specified by the user. Hence, to complete the path creation, the user has to modify the position and/or orientation of the VP sending it back again to the remote PC (Fig. 4c and Fig. 4d illustrate a wrong VP and the same VP after having being corrected by the user, respectively).

Although, at the current stage, the system does not provide a real-time objects tracking mechanism, the AR interface provides a collision detection capability: if the spline or one of the VP intersect an obstacle, their color is changed



(a) (b)

Fig. 5. (a) The end-effector path collides with an obstacle. Its color is changed to red to highlight the collision. (b) The collision with the visualization of the end-effector disabled. (Color figure online)

to red, highlighting a possible collision between the object and the real end-effector. To simulate the object detection process, the real objects have been placed at a known location with respect to the image target. Then, their virtual counterparts have been positioned at the same location, allowing to perform the collision detection mechanism (Fig. 5).

Finally, the user can delete one VP at a time by selecting it and by pressing the “Delete” button positioned at the bottom right-side of the user interface. Furthermore, the users can also delete all the VPs at the same time by pressing the “Clear yellow pts” button placed at the bottom left-side of the user interface. A video showing the main functionalities of the proposed AR interface can be found at[1].

5 The User Study

In order to assess the effectiveness of the proposed solution, twelve users have been involved in a user study at Politecnico di Torino university. The study has been divided into two different tasks: (i) 2D translational and rotational (TR_2D) and (ii) 3D translational and rotational (TR_3D) tasks. Generally, the tasks required the users to create a virtual path in the real environment, under some specific conditions. The labels *2D* and *3D* refer to whether the task was constrained on the plane on which the e.DO manipulator has been placed or to the 3D space surrounding the robot, respectively. The label *TR* indicates that the path should be generated using translational and rotational (TR) actions (see Sect. 4.2).

Each task required the creation of one single path, composed by two distinct VPs. In order to assess the accuracy of the proposed AR interface, the VPs

---

[1] https://www.youtube.com/watch?v=xvJhPJ50-xg.

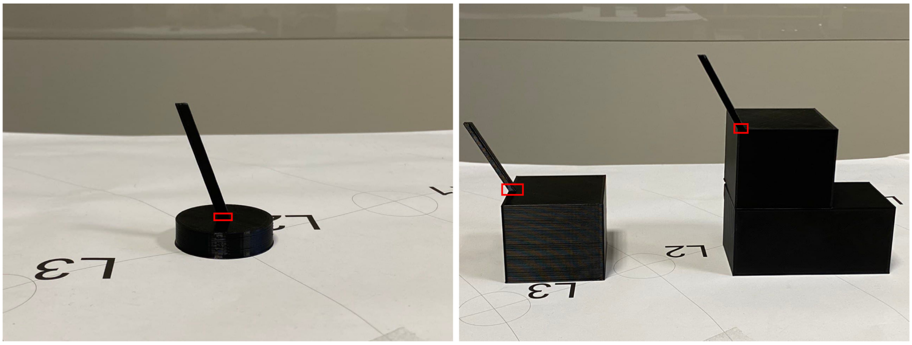
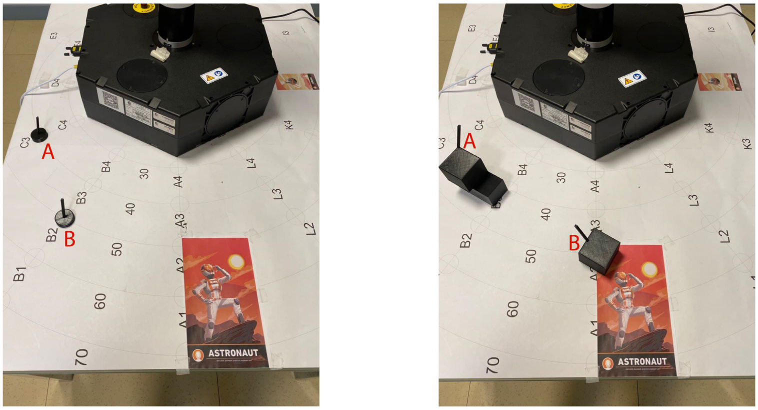

(a) (b)

Fig. 6. (a)–(b) The printed models used for the TR_2D and TR_3D tasks, respectively. The red rectangles highlight the position that the end-effector should reach. The stick inclination represents the orientation that the end-effector should have at the specific position. (Color figure online)

had to be placed in a set of known positions with pre-defined orientations (the ground truth points, GTPs). The GTPs thus represented both the position and orientation that the real end-effector should had at the specific positions. The GTPs were represented by some 3D printed models with different shapes and sizes. The models represented in Fig. 6a have been used for the TR_2D task, whereas the models shown in Fig. 6b have been employed for the TR_3D task. The purpose of the vertical stick was twofold: (i) the tip of the stick (highlighted in red in Fig. 6) indicated to the users the position that the end-effector should reach whereas (ii) the stick inclination represented the orientation that the end-effector should have at the specific position. The GTPs have been positioned in pre-defined locations with respect to the image target, thus allowing to compute the GTPs positions and orientations.

Both objective and subjective parameters have been collected and analyzed. Specifically, the objective consisted of: (i) the time required to complete the task, (ii) the number of user's errors, (iii) the number of touch interactions and, (iv) the positional and rotational differences between the VPs and GTPs. The task time represents the time between the starting of a specific task and the positioning of the two VPs (an external operator was monitoring and recording the users' actions). Every time the users were pleased with their positioning of the VPs, the external operator stopped the time. It is worth noticing that although users could be accurate in placing the VPs, the accuracy could not be enough for the IK solver running at the PC side. In this case, the IK solver would send back an error message, highlighting the wrong VP (see Sect. 4.2). Hence, each user could try the creation of a path for at maximum three times and a global time value was computed as the sum of the time values required at each trial (same procedure applied for the touch values). Referring to the VP-GTP differences, two different types of accuracy have been computed, represented by the positional and rotational errors (PE and RE, respectively). PE has been computed following the ISO 9283 standard [44], that is, the square root of the distance between the VP and GTP X, Y and Z coordinates. Similarly, RE has been determined by calculating the distance angle between the VP and GTP rotations. Considering the subjective parameters, four different types of data have been collected: (i) user general data (UGD), (ii) interface usability (I_USA), (iii) interface workload (I_WOR) and, (iv) a global score indicating whether the interface was adequate or not to effectively complete the tasks (GLOS). UGD consisted of sex, age and the level of previous experience with AR and robotics. The I_USA and I_WOR have been assessed using the System Usability Scale [45] and the NASA-TLX [46] questionnaires, respectively. GLOS has been determined using the Single Ease Questionnaire [47].



(a) (b)

Fig. 7. (a)–(b) The TR_2D and TR_3D tasks, respectively. The users had to generate a virtual path passing through the A and B points.

The overall user study procedure can be summarized as follows: firstly, the user has been introduced to the experiment by letting them try the interface in several trial scenarios. The scenarios have been designed to encourage the users to apply the translational and rotational actions, thus improving the user knowledge of the interface. Moreover, they have been explained the purpose of the 3D printed models, with particular attention for the model sticks. Then, the user filled the UGD section and started the sequence of tasks (TR_2D-TR_3D, refer to Fig. 7). The VPs were already instantiated in the virtual scene at the image target coordinates, thus guaranteeing the same starting conditions across all the users. Then, the user started a task trying to generate a path as quickly as possible. The starting joint robot configuration has been kept the same across all the tasks with the joint values equal to 0°. In case the user generated a virtual path feasible for the real manipulator, he/she had the possibility to send it to the real robot to visualize the real robot motion. After having completed all the tasks, the user had to fill the I_USA, I_WOR and GLOS sections of the questionnaire. He/she could also verbally report to the external operator additional comments regarding the overall experience.

In the next section, the related results are presented and discussed.

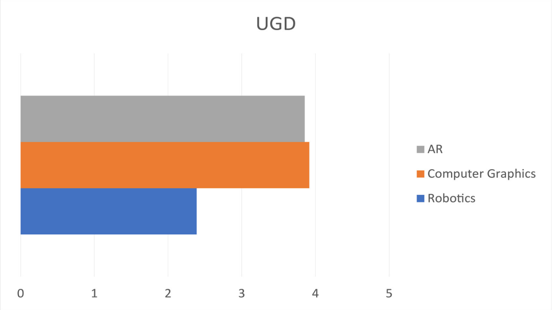


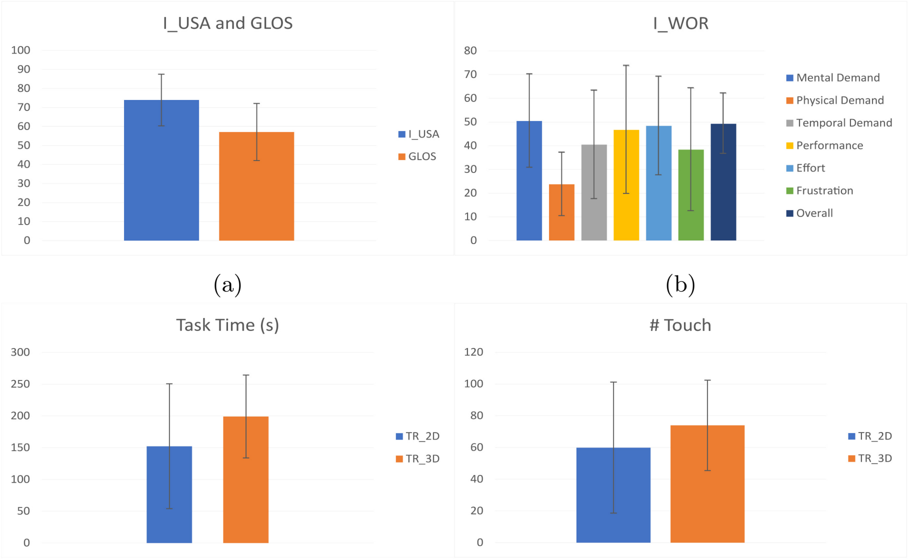
Fig. 8. The UGD outcomes ranging from 0 (not familiar) to 5 (extremely familiar).

6 Results and Discussion

The users were all male with an average age of 27 years old. The users were found to be familiar with AR and the computer graphic context whereas they had moderate knowledge of the robotic domain (Fig. 8).

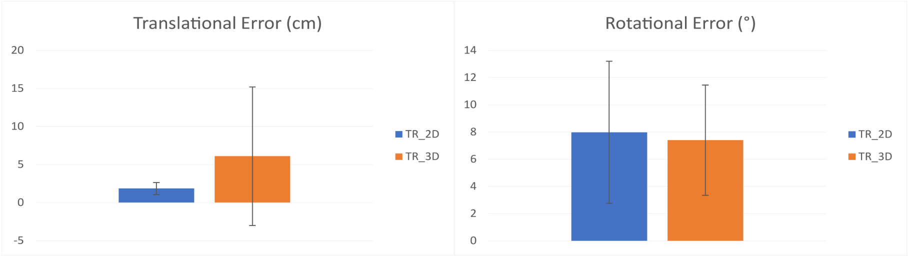
The I_USA score has been determined by computing the average value that has been used to compute the usability score using the approach presented in [45], whereas the GLOS average value has been normalized in the 1–100 interval. The USA score indicates that the AR interface has been deemed sufficiently appropriate to generate the virtual robot path, although there seems to be space for future improvements. The high standard deviation value of the GLOS outcome suggests a non uniform users' opinion regarding the effectiveness of the AR interface for the evaluated tasks. Regarding the I_WOR scores, the relatively high mental and effort values indicate that the users had difficulty in reasoning about the correct VP configuration. Some users reported that the manipulation of the VP rotation was quite hard, whereas the VP translation methodology was appropriate for the considered tasks. Figure 9a and Fig. 9b illustrate the I_USA, GLOS and I_WOR results, respectively.

Figures 9c–9d–9e–9f show the objective results. Independently of the data, the collected values have been summed up across all the users computing the corresponding average values. Unsurprisingly, both the task time (Fig. 9c) and number of touch interaction (Fig. 9d) values show higher scores for the TR_3D task with respect to the TR_2D one. Having to manage an extra dimension, the users seem to have encountered more difficulties in the 3D task then the 2D one.



(a) (b)

(c) Task time. (d)



(e) (f)

Fig. 9. (a–b) The usability and workload results, respectively. (c–d) The task time and number of touch interactions, respectively. (e–f) The translational and rotational errors, respectively.

The results are partially confirmed by the translational errors (Fig. 9e), whereas the rotational ones (Fig. 9f) appear to be consistent across the task modalities. It is reasonable to assume that there are no significant differences between rotating a VP positioned on a 2D plane and one placed on a 3D model, at a different height with respect to the base plane.

7 Conclusion

This paper proposed the evaluation of an AR handheld interface to program a real robot manipulator. The interface allows the users to generate and visualize virtual robot paths directly in the real environment. By pre-visualizing the robot path, the users can understand if it is adequate for the considered task.

The interface has been evaluated through a series of user tests. The novelty of the work consists into assessing both 2D and 3D tasks, that have not been previously considered in the related state of the art. The main results suggest that the 3D tasks are far more compelling than the 2D ones. Furthermore, the users deemed the interface as fairly appropriate for the evaluated scenario. This outcome suggests that there is still room for improving the AR interfaces in this peculiar domain.

Future works will compare the handheld interface with wearable ones to verify whether a changing in the interaction and visualization paradigms affects the generation of the virtual robot paths.

References

1. Lasi, H., Fettke, P., Kemper, H., Feld, T., Hoffmann, M.: Industry 4.0. Bus. Inf. Syst. Eng. **6**, 239–242 (2014)
2. Hofmann, E., Rüsch, M.: Industry 4.0 and the current status as well as future prospects on logistics. Comput. Ind. **89**, 23–34 (2017)
3. Kagermann, H., Wahlster, W., Helbig, J.: Others recommendations for implementing the strategic initiative Industrie 4.0: final report of the Industrie 4.0 working group. Forschungsunion, Berlin, Germany (2013)
4. Erboz, G.: How to define Industry 4.0: the main pillars of Industry 4.0 (2017)
5. De Pace, F., Manuri, F., Sanna, A.: Augmented reality in Industry 4.0. Am. J. Comput. Sci. Inf. Technol. **6**, 17 (2018)
6. Vaidya, S., Ambad, P., Bhosle, S.: Industry 4.0 - a glimpse. Procedia Manuf. **20**, 233–238 (2018)
7. De Pace, F., Manuri, F., Sanna, A., Fornaro, C.: A systematic review of augmented reality interfaces for collaborative industrial robots. Comput. Ind. Eng. **149**, 106806 (2020)
8. Robotics, I.: Top trends robotics 2020. https://ifr.org/ifr-press-releases/news/top-trends-robotics-2020
9. Gaz, C., Magrini, E., De Luca, A.: A model-based residual approach for human-robot collaboration during manual polishing operations. Mechatronics **55**, 234–247 (2018)
10. Krupke, D., Steinicke, F., Lubos, P., Jonetzko, Y., Görner, M., Zhang, J.: Comparison of multimodal heading and pointing gestures for co-located mixed reality human-robot interaction. In: 2018 IEEE/RSJ International Conference On Intelligent Robots And Systems (IROS), pp. 1–9 (2018)
11. Araque, D., Díaz, R., Pérez-Gutiérrez, B., Uribe, A.: Augmented reality motion-based robotics off-line programming. In: 2011 IEEE Virtual Reality Conference, pp. 191–192 (2011)
12. Sita, E., Studley, M., Dailami, F., Pipe, A., Thomessen, T.: Towards multimodal interactions: robot jogging in mixed reality. In: Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology, pp. 1–2 (2017)
13. Frank, J., Moorhead, M., Kapila, V.: Realizing mixed-reality environments with tablets for intuitive human-robot collaboration for object manipulation tasks. In: 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 302–307 (2016)
14. Frank, J., Moorhead, M., Kapila, V.: Mobile mixed-reality interfaces that enhance human-robot interaction in shared spaces. Front. Robot. AI **4**, 20 (2017)

15. Lambrecht, J., Krüger, J.: Spatial programming for industrial robots based on gestures and augmented reality. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 466–472 (2012)
16. Lambrecht, J., Kleinsorge, M., Rosenstrauch, M., Krüger, J.: Spatial programming for industrial robots through task demonstration. Int. J. Adv. Rob. Syst. **10**, 254 (2013)
17. Hügle, J., Lambrecht, J., Krüger, J.: An integrated approach for industrial robot control and programming combining haptic and non-haptic gestures. In: 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 851–857 (2017)
18. Akgun, B., Cakmak, M., Yoo, J., Thomaz, A.: Trajectories and keyframes for kinesthetic teaching: a human-robot interaction perspective. In: Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction, pp. 391–398 (2012)
19. Gradmann, M., Orendt, E., Schmidt, E., Schweizer, S., Henrich, D.: Augmented reality robot operation interface with Google Tango. In: 50th International Symposium on Robotics, ISR 2018, pp. 1–8 (2018)
20. Chacko, S., Kapila, V.: An augmented reality interface for human-robot interaction in unconstrained environments. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3222–3228 (2019)
21. Kapinus, M., Beran, V., Materna, Z., Bambušek, D.: Spatially situated end-user robot programming in augmented reality. In: 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pp. 1–8 (2019)
22. Blankemeyer, S., Wiemann, R., Posniak, L., Pregizer, C., Raatz, A.: Intuitive robot programming using augmented reality. Procedia CIRP **76**, 155–160 (2018)
23. Chong, J., Ong, S., Nee, A., Youcef-Youmi, K.: Robot programming using augmented reality: an interactive method for planning collision-free paths. Robot. Comput. Integr. Manuf. **25**, 689–701 (2009)
24. Fang, H., Ong, S., Nee, A.: Interactive robot trajectory planning and simulation using augmented reality. Robot. Comput. Integr. Manuf. **28**, 227–237 (2012)
25. Fang, H., Ong, S., Nee, A.: Robot path and end-effector orientation planning using augmented reality. Procedia CIRP **3**, 191–196 (2012)
26. Fang, H.C., Ong, S.K., Nee, A.Y.C.: Novel AR-based interface for human-robot interaction and visualization. Adv. Manuf. **2**(4), 275–288 (2014). https://doi.org/10.1007/s40436-014-0087-9
27. Pai, Y., Yap, H., Singh, R.: Augmented reality-based programming, planning and simulation of a robotic work cell. Proc. Inst. Mech. Eng. Part B J. Eng. Manuf. **229**, 1029–1045 (2015)
28. Zaeh, M., Vogl, W.: Interactive laser-projection for programming industrial robots. In: 2006 IEEE/ACM International Symposium on Mixed and Augmented Reality, pp. 125–128 (2006)
29. Veiga, G., Malaca, P., Cancela, R.: Interactive industrial robot programming for the ceramic industry. Int. J. Adv. Robot. Syst. **10**, 1 (2013)
30. Reinhart, G., Vogl, W., Kresse, I.: A projection-based user interface for industrial robots. In: 2007 IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems, pp. 67–71 (2007)
31. Tanzini, M., Tripicchio, P., Ruffaldi, E., Galgani, G., Lutzemberger, G., Avizzano, C.: A novel human-machine interface for working machines operation. In: 2013 IEEE RO-MAN, pp. 744–750 (2013)

32. Kyjanek, O., Al Bahar, B., Vasey, L., Wannemacher, B., Menges, A.: Implementation of an augmented reality AR workflow for human robot collaboration in timber prefabrication. In: Proceedings of the 36th International Symposium on Automation and Robotics in Construction (2019)
33. Quintero, C., Li, S., Pan, M., Chan, W., Loos, H., Croft, E.: Robot programming through augmented trajectories in augmented reality. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1838–1844 (2018)
34. Ong, S., Yew, A., Thanigaivel, N., Nee, A.: Augmented reality-assisted robot programming system for industrial applications. Robot. Comput. Integr. Manuf. **61**, 101820 (2020)
35. Guan, Z., Liu, Y., Li, Y., Hong, X., Hu, B., Xu, C.: A novel robot teaching system based on augmented reality. In: 2019 International Conference on Image and Video Processing, and Artificial Intelligence, vol. 11321, id. 113211D (2019)
36. Manring, L., et al.: Augmented reality for interactive robot control. Spec. Top. Struct. Dyn. Exp. Tech. **5**, 11–18 (2020)
37. Chacko, S., Granado, A., Kapila, V.: An augmented reality framework for robotic tool-path teaching. Procedia CIRP **93**, 1218–1223 (2020)
38. unity.com: Unity. https://unity.com
39. developer.vuforia.com: Vuforia SDK. https://developer.vuforia.com/downloads/sdk
40. releases.ubuntu.com: Ubuntu 20.04.2.0. http://www.releases.ubuntu.com/20.04/
41. wiki.ros.org: ROS Melodic. https://wiki.ros.org/melodic
42. comau.com: COMAU e.DO. https://www.comau.com/en/our-competences/robotics/edo
43. moveit.ros.org: MoveIt. https://moveit.ros.org/
44. Iso.com SO 9283:1998: Manipulating industrial robots - performance criteria and related test methods. https://www.iso.org/standard/22244.html
45. Brooke, J., et al.: SUS: a ‘quick and dirty’ usability scale. In: Usability Evaluation In Industry, vol. 189, pp. 4–7 (1996)
46. Hart, S., Staveland, L.: Development of NASA-TLX (task load index): results of empirical and theoretical research. Adv. Psychol. **52**, 139–183 (1988)
47. tinyurl.com: Measuring task usability: the single ease question (SEQ). https://tinyurl.com/24p2x2dd