www.sensorsportal.com

Sensors & Transducers
Published by IFSA Publishing, S. L., 2023
http://www.sensorsportal.com

Learning from Demonstration and Safe Cobotics Using Digital Twins

[1*] A. RIZZOTTI-KADDOURI, [1] L. JEANNERET, [1] B. STUDER, [1] J. BRACAMONTE, [1] N. OUERHANI and [2] M. KUNZE
[1] School of Engineering Neuchâtel, University of Applied Sciences and Arts Western Switzerland, HES-SO/HE-ARC, Espace de l'Europe 11, 2000 Neuchâtel, Switzerland
[2] School of Engineering and Management Vaud, University of Applied Sciences and Arts Western Switzerland, HES-SO/HEIG-VD, Route de Cheseaux 1, 1401 Yverdon-les-Bains, Switzerland
[1] Tel.: +41 32 930 22 94, fax: +41 32 930 22 94
E-mail: aicha.rizzotti@he-arc.ch

Received: 14 April 2023 /Accepted: 30 May 2023 /Published: 26 June 2023

Abstract: The use of collaborative robots, or cobots, is nowadays continually increasing, especially in the small- and medium-sized manufacturing sector. For each particular use case, the integration and deployment of a cobot into a collaborative workspace faces a certain number of challenges. Programming industrial robots, for example, can be a relatively complex and time-consuming task. In this paper we report an accurate method to robot programming by using an optimized "learning from demonstration" technique. The operator/programmer performs in real-time the corresponding task to be automatized, and by means of a tracker sensor the programmer's motions are captured and transmitted to the robot; the robot registers the trajectories and is now able to reproduce the human movements with high accuracy. Another fundamental issue for cobot deployment is safety. In this paper, we also present a virtual/augmented reality (VR/AR) environment to facilitate the design and operation of cobots in order to maximize human safety. The virtual reality environment operates as an aide tool during the design phase. The human operator and the robot's digital twin work side-by-side while executing a collaborative task in a virtual reality space. Their movements are controlled and registered, and after a given period of test time, the data is analyzed to suggest modifications to ensure a safe workspace (collision free) and to increase productivity. For the regular real-time cobot operation, an augmented reality environment was developed, again, with the purpose of assuring a safe human-robot collaboration. The augmented reality environment keeps tracking permanently the cobot and the human manipulations. This system produces audio and visual alarm signals in unsafe situations and is also able to take actions, such as slowing down or stopping the robot, to preserve the physical integrity of the human operator.

Keywords: Cobot, Human-robot collaboration, Virtual- and augmented reality, Learning from demonstration, Machine learning, Digital twin, Sensors, Smart glasses.

1. Introduction

Collaborative robots represent an effective strategy for promoting human-machine interaction to increase productivity in the manufacturing room. Cobots relieve humans of repetitive, dangerous, non-ergonomic or heavy-load tasks. The goal is that cobots and humans collaborate side-by-side sharing the same workspace in a safe and efficient manner.

Depending on the use case, the design and deployment of a cobot faces a certain number of challenges. In this paper we address some of those challenges.

This paper is an extended version of the conference publication [1] in which we presented a solution to the complex task of programming robots. In this paper we report the general context and additional research on the development of both a virtual- reality and an augmented-reality environment. The objective of these two systems is to design and operate safer workspaces (e.g., zero collisions) in industrial human-robot collaboration. The virtual reality environment is used specifically for the human-cobot workspace design while the augmented reality environment is rather used during the operation phase (real-time, real-world human-robot collaboration).

2. Related Works

The LfD (Learning from Demonstration) robot-programming approach has attracted a lot of interest in the field of human-robot interaction [2, 3]. This topic encompasses several disciplines and scientific fields.

In general, there are two main approaches for LfD. The first one is based on observational learning which usually exploits a vision system for the perception of movements and gestures, by using for example, cameras. The second approach is based on kinesthetic guidance which refers to the manual movements of the robot by interaction through haptic sensors.

It should be noted that the second approach simplifies the corresponding learning task, but the movements remain spatially limited. Moreover, it is not possible to perform kinesthetic guidance on all types of robots. Therefore, in this study we favored the vision-based learning approach. We note that there are several important phases in this workflow, namely:

data capture and fusion > learning phase > reproduction by the robot of the learned movement or action.

Relevant existing work in this area is reported below.

In the framework of the European project PRACE (Productive Robot ApprentiCE), whose aim is the development of a mobile robotic platform for the automation of assembly operations, a system based on several Kinect-type cameras to calculate 3D positions has been investigated. In this work [4], the author adopted a top-down approach: first, an estimation of the positions was made, and then, a refinement of the data captured by each sensor was performed.

In the LfD domain, there are other approaches [5] that are based on the fusion of data recorded with gloves and video cameras. The work in [6] up to a certain point inspired our approach of fusing data from Microsoft's Kinect V2 to get the 3D data in skeletal form and supplementing it with data from Intel's RealSense sensor to refine the depth.

One example of this approach is also reported in [7], where the robot must perform assembly tasks in the industrial Peg-in-Hole (PiH) domain. Here, a learning phase and a reproduction phase reinforced by a kinesthetic guidance phase are listed. With a camera on the robot's wrist, the object is detected, located and captured. The objects are labelled, and the detection algorithms are based on conventional computer vision algorithms such as SIFT (Scale-Invariant Feature Transform) and KNN (K-Nearest-Neighbors) for classification.

We note that in [8] the human arm and the robot arm are physically attached for learning. The trajectory of each action is followed by the robot arm giving a representation of the trajectory in space. For the demonstration, the authors used the 3D coordinates of the three joints (shoulder, elbow, and wrist) for the tracking. They also used the procedure named Gaussian Process Latent Variable Model (GPLVM) and RANSAC (RANdom SAmple Consensus) to map the motion behavior.

In [18] the imitation is done by decomposing the main task into hierarchical subtasks, based on an RNN neural network to predict the next task to be performed. This will of course depend on the observed input of the current state with the control of a closed loop.

The work reported in [19] is carried out by using the ABB YuMi robot which is the same robot we used to develop our approach. The aim of the experiment was to test the effects of reusing non-expert programming parameters and skills for assembly tasks on industrial robots.

Our approach of course leveraged the results from previously reported studies. Our objective nonetheless was to increase accuracy (a recurrent problem in the cited works) either by using multi-channel vision systems for the data captures (unlike [3, 8]), or to obtain finer resolution movements (unlike [4, 5, 6]).

3. Demonstration-based Robot Learning

Efficient robot programming is one of the challenges faced during the deployment of cobots. Demonstration-based robot learning [9] is an active research area which studies robot programming. In this area, capturing and replicating the motion with high accuracy remains a recurrent issue. Our approach [1] to tackle the accuracy problem was to use a system composed of multiple sensors [10] and then to extract a trajectory from the motions and the (programmer's) fingers' positions using machine learning.

We tested different sensors including gloves and cameras. Due to accuracy and occlusion problems, we opted for a solution which uses an HTC Vive Tracker sensor [17], which is indeed accurate enough for robotic applications [11]. In our workflow, the HTC VIVE Tracker sensor is attached to the application-dependent tool which is used by the operator "to demonstrate" the motion. The Tracker sensor can then track the movement of the tool in three dimensions by locating its position relative to a VIVE Base Station.

Since the tool position and rotation is directly tracked, it is not necessary to interpret (or to process) the intent and gestures of the user, which would have required more complex machine learning models. Moreover, our solution is coupled with AR smart glasses which allows the user to have immediate feedback. The used AR glasses (the Microsoft HoloLens 2) also provide tools to correct, in an intuitive way, possible recording errors.

3.1. Safe Human-robot Collaboration Testing

Robotic Task Planning, which is a sub-category of automated planning and scheduling, aims at solving complex robot use-case scenarios [12]. While some problem solvers such as STRIPS [13] can integrate human safety (as described in [14]) we cannot integrate such a system when a task is learned from demonstration. As an alternative, we approached this issue by testing the learned tasks in a high-fidelity virtual reality environment, where the human operator does not risk any harm. In our safe testing environment multiple digital twins can be placed in the scene along with a real human operator. The digital twins can be connected either to an alternative real simulator or to real robots located in a separate room.

4. VR and AR Environments for Safe Cobotics

Safety is a non-negotiable requirement for any human-robot collaboration in the workplace. One of the main challenges for a wide deployment of robots/cobots is related to safety issues. To unlock the full potential of collaborative robotics in industry and society, human safety must be guaranteed. At the same time, investment in the safety of human-robot collaboration must not reduce the promised return on investment.

Below we report the current status of the development of a VR/AR system intended for achieving safer workspaces (e.g., zero collisions) and increased productivity in industrial human-robot collaborations.

During the real-time cobot operation phase, the same digital twin used in Virtual Reality is used in Augmented Reality with AR smart glasses (in this case, the Microsoft HoloLens 2), while simultaneously calculating and displaying other kinds of metrics. One, for example, is the *Safety Score* metric. This score indicates to the operator the degree of safety of the current situation (or position). If the situation is deemed too risky, or when the operator crosses a dangerous zone, the smart glasses can send a signal to slow down or to stop the robot.

4.1. System Architecture

The system is composed of two main sub-systems as can be seen in Fig. 1. Each of the two main parts of this figure can operate independently on its own. They communicate together through robots and simulators by sharing programmed trajectories.

The "Common Digital Twin Library" component (on the right side of Fig. 1) provides a generic way to define robots for Virtual Reality and Augmented Reality called *Robot Components.* Robot Components allow us to define an existing robot or to prototype a new one by using small re-usable parts.

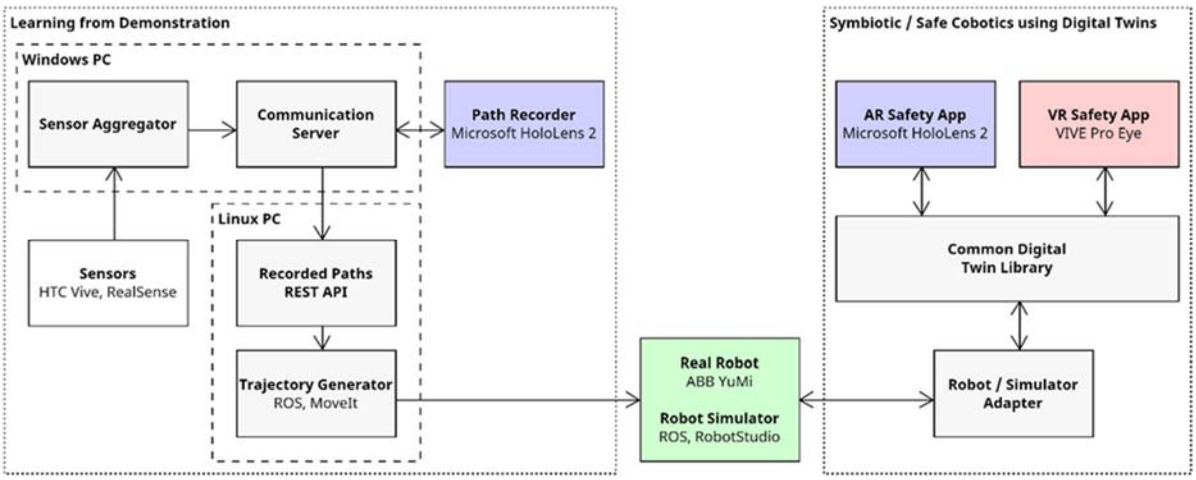

Fig. 1. Learning from demonstration and safe cobotics using digital twins.

Robot Components can be very specific and accurate (e.g., an accurate 3D model of an existing robot) or very generic and re-usable (e.g., a robotic joint, a 3D collider, a generic gripper, etc.). They can then be combined, using a parent/child system to define a fully working robot. This allows us to add support for many robots' features with minimal effort, as long as the simulator or the real robot provides an API for it.

Robot Components can communicate to real robots and simulators through the "Robot / Simulator Adapter" component, which is a custom NodeJS server with an adapter system.

Plugins can add new adapters, which permits to communicate with both real robots and simulators. Robot Components and the plugin system make it possible and easy to add support for new robots if they have a public API.

Robots are *stored* in a modeled room (which represents a real-life industrial room), which can then be loaded in Virtual Reality to test the scenario, or in Augmented Reality (without furniture) to show to the operator where the robots are. Fig. 2 shows the class diagram of how robots are stored in a room.

Our *Learning from Demonstration* system (on the left in Fig. 1) is composed of multiples modules and several physical sensors. Two *sensor aggregator applications* retrieve and merge data from the sensors. A communication bus exchanges data between the HoloLens 2 module, the web server, and the sensor aggregators. The server module stores the recorded trajectories. It is then accessed by the ROS (Robot Operating System) module, which converts the operator trajectories into robot trajectories, before sending them to the robot. The sensor aggregator is built in a way that makes it easy to add new sensors, by allowing each sensor to validate or improve the accuracy of previous sensors.

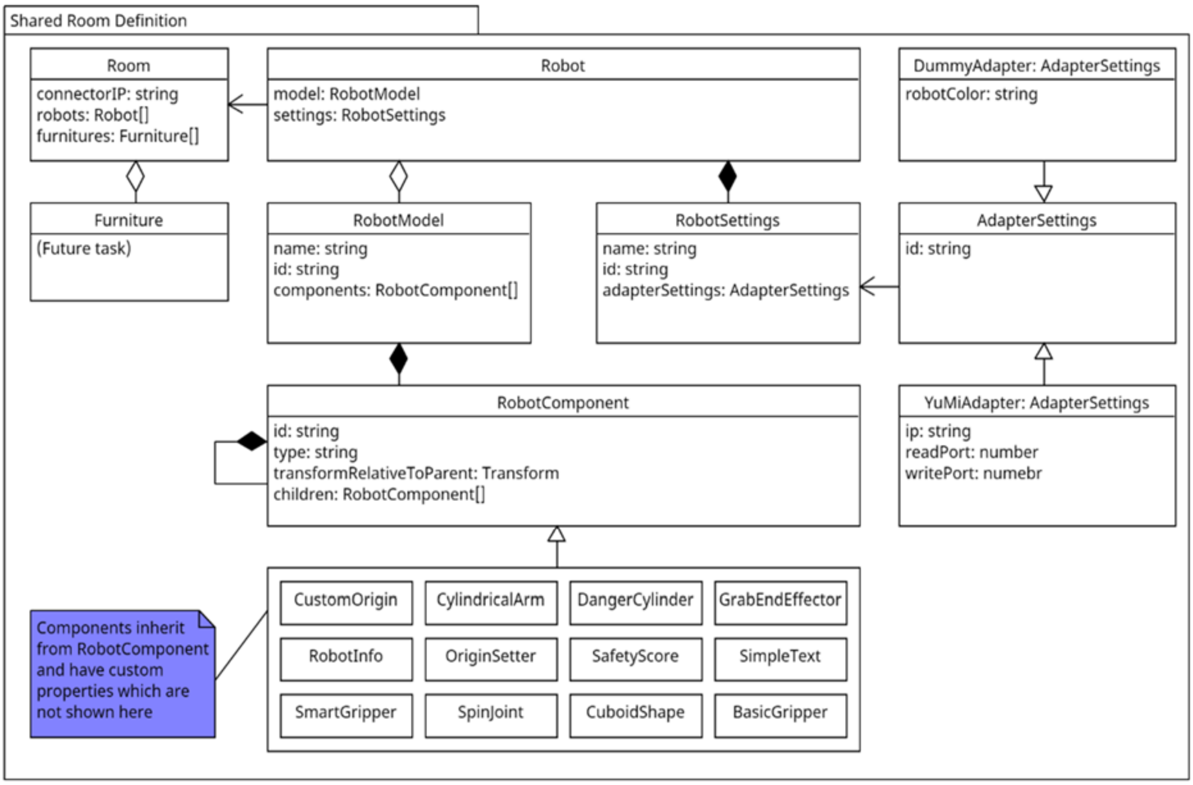

Fig. 2. Room definition class diagram used in the VR and AR environments.

4.2. Sensors

4.2.1. Leap Motion and Deep Learning

The goal of this part is to set up a system for the detection and classification of gestures performed by a human hand. The acquisition of gesture data is done using a *Leap Motion*, and the analysis and classification with a deep learning model (see Fig. 3). The process is carried out in three parts: The first one is the data acquisition, the second one is the creation and training of an MLP (Multilayer Perceptron) neural network, and the third part consists in importing the previously created model and making predictions on the fly with it. The features used are the relative distances between the fingers, which allows reducing the neural network complexity.

The main objectives for this part were successfully attained. For example, the "pinch" and "thumb up" hand gestures were correctly recognized. Other hand signs can of course be added at the expense of longer training and slightly lower performance. One recurrent issue we had was the misinterpretation of gestures that are differentiated only by hand rotation. A straightforward solution to this problem and/or a possible improvement will be the inclusion of hand orientation as an additional feature for the neural networks training.

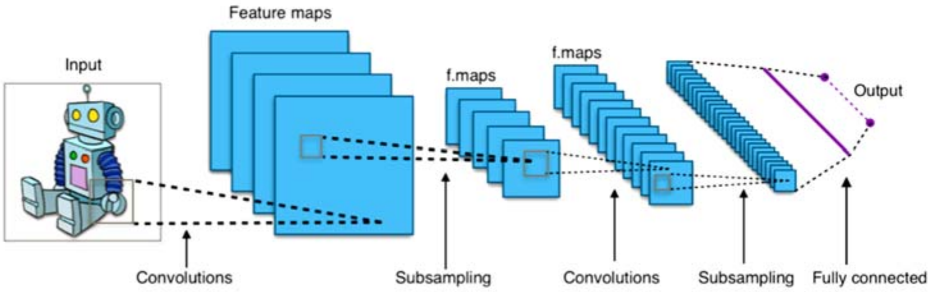

Fig. 3. Classification with the deep learning model.

4.2.2. Camera

In order to increase accuracy, we opted to use the RealSense D435 sensor (depth camera), which allows the detection of objects with high precision. Subsequently we merged the captured information with the Trackers' data. The depth camera is placed above the workspace. An algorithm detects the plane of the table and then a threshold is applied to the depth of each pixel. This allows to detect items dimensions with accuracy (±3 mm delta error) on different types of boxes. See Fig. 4.

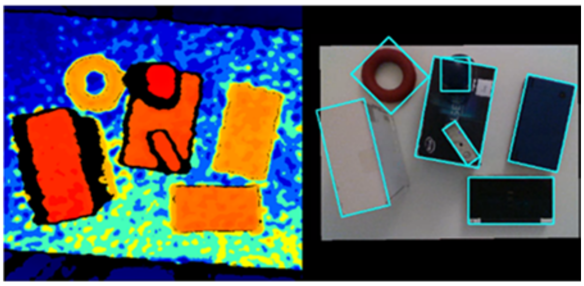

Fig. 4. Camera sensor and boundary detection algorithm.

4.2.3. Hand Tracking with Gloves

Multiple hand tracking sensors were tested, including the Hi5 Glove and the Senso Gloves before settling for the HTC VIVE Tracker. The main disadvantages of the non-selected ones were that they could be cumbersome to wear and that they featured a lower tracking accuracy. Additionally, their main purpose is to track the fingers and hand position of the operator, and for our application, it was not always possible to convert (post-process) the obtained data to accurately describe the motion of the *tool*.

To solve this in a pragmatic way, we decided to attach the tool to the Tracker. Since we could not use the gripper of the robot directly, we 3D-printed a "pen" representing a gluing tool (see use-case in the next section). However, in future work, it will be possible to print custom tips, either to attach existing tools, or to print "smart" tools. For example, a gripper with a small electronic circuit could automatically track the opening of the clamp during the recording, which would be feasible via the HTC VIVE Trackers' programmable pins.

4.2.4. HTC VIVE Tracker and AR Smart Glasses

A pair of Augmented-Reality/Mixed-Reality glasses were used in combination with two HTC VIVE Trackers. The selected AR/MR glasses were the Microsoft Hololens 2 glasses [20]. The system architecture is, all the same, open to the use of other types of glasses or models from other manufacturers.

One of the HTC VIVE trackers was used to track the tool and the other to define the origin. A quick response QR code (see Fig. 5) was used to synchronize the origins of the Tracker and the HoloLens 2. The operator can view the limits of the workspace through the AR/MR smart glasses and use controls to start and stop recording (Fig. 5).

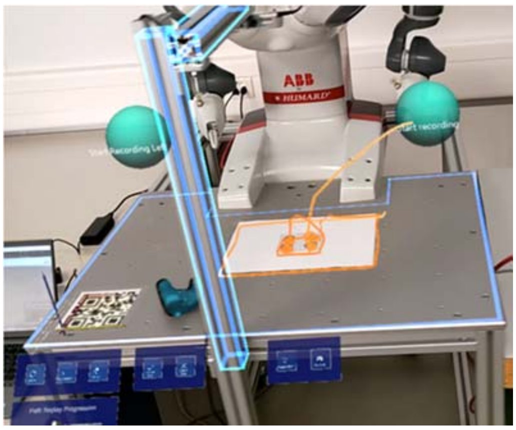

Fig. 5. AR smart glasses screen capture.

Once the task is recorded, it can be uploaded to a web server for future usage. The communication between the web server, the HoloLens 2, and the Trackers is done in a modular way.

The accuracy of the Tracker with the pen was measured by touching with the tip of the pen at specific (known distance) points drawn on a sheet of paper. The results of the positions are summarized below in Fig. 6.

x	y	z	x	y	z	x	y	z
Expected			Measured			Absolute error		
0	0	0	1	1	1	1	1	1
0	0	53	0	0	54	0	0	1
0	0	143	3	8	145	3	8	2
(More lines truncated...)								
HTC Tracker			Average :			1.875 [mm]		
			Max :			8 [mm]		

Fig. 6. Accuracy of the HTC Vive Tracker.

The use case for our demonstration was a "gluing task", which has the following sequence: the operator picks up an item, moves it, applies glue on it, and then glues the other half together. Fig. 7 shows the operator during the recording of the trajectory, which is displayed in orange color. While this figure displays a 2D path, the path recorded is indeed three-dimensional.

For the reported use case, an ABB YuMi robot was used. This robot has two arms. The operator can decide which arm is used while *in teaching mode* by simply touching the arm of the robot with the pen. Technically, two Trackers could be used to record trajectories for both arms at the same time.

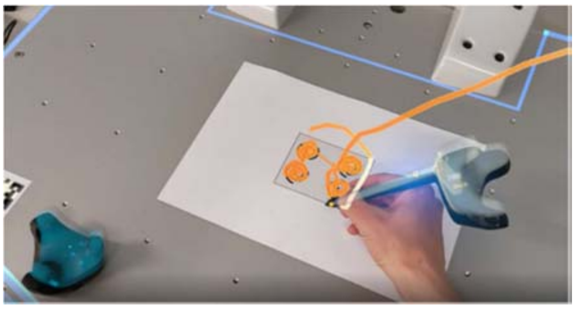

Fig. 7. Recording the trajectory.

4.3. VR environment for a Safe and Efficient Human-robot Workspace

Once the trajectories are recorded using the method described in the previous sections, they are validated in a Virtual Reality environment. Contrary to traditional approaches that would try to test the trajectories in real conditions with a physically present robot, we opted to equip the operator with an HTC Vive Pro eye head mounted display [15] and use Unity's Virtual Reality features [16] coupled with digital twins. Such a system reduces the injury risks of the human operator to practically zero when testing new trajectories. This riskless procedure also minimizes the overall testing time. Furthermore, a virtual environment permits to create new testing scenarios that would be complex and/or more expensive to set up in real life. The VR system also facilitates the testing of different or alternative configuration scenarios in search of increasing productivity.

The following paragraphs describe the operation of the VR system in more detail:

1) The operator can (virtually) observe both the robot and the surrounding environment.
2) The operator can then at real-time speed interact with the robot and the environment as in a classic Virtual Reality application. The simulation is as close as possible to the real-world case from the operator's standpoint. A connection with the Robot Adapter allows the robot to mimic real-life robot trajectories.
3) For post-analysis and optimization purposes, the VR system offers the capability of recording the scenario and all the involved actors (e.g., the robot(s), the operator in the room).
4) Diagnostic tools and metrics are also constantly shown to the human operator. For example, we can display the minimum distance between the operator and the robot(s) for every timestamp during the execution of the use case scenario; or the number of occurred human-robot (virtual) collisions. Other metrics can be easily implemented and added in graphs to the displayed information. Fig. 8 shows the diagnostic view for a very simple use-case example. On the top left part of the display the number of collisions is reported. On the right part, the minimum distances and robot speed plots are displayed.

Fig. 8. Diagnostic view in a simple use case.

4.3.1. Testing the System: Results

In order to test our system on a practical application, a pick-and-place (robot-executed) task with an intermediate assembly task (human-executed) was successfully implemented. We used multiple sensors and a real (not a virtual one) ABB YuMi robot.

During the tests, it was confirmed that the hand-tracking accuracy (by using the Microsoft HoloLens 2) was high enough to control the ABB YuMi robot.
Our plugin-based system allows us to quickly add new robots and simulators. For example, we exercised this feature by providing multiple plugins for multiple entities: a) a plugin that can communicate with a real ABB YuMi, b) another plugin with a simulated ABB robot, and c) a third plugin with a modeled "dummy" (robot) which was used to simply test the feasibility with any prototyped robot.

Fig. 9. Screen capture of the AR smart glasses during regular human-robot operation. The human operator can visualize in real-time the danger zone, the safety score, and other valuable information.

The developed modular Robot Components system was used to define an ABB Robot, which can be animated using data from real robots and simulators. These digital twins can be viewed in Virtual Reality and in Augmented Reality.

The implemented solution allows managing tasks on a running robot. When the robot is stopped, it can be controlled in an intuitive manner.

Safety features were implemented. For example, displaying a danger zone (the circular yellow strips in Fig. 9); or slowing or stopping the robot depending on the value of the *safety score* or on the distance of the operator to the robot.

Furthermore, the robots can be visualized in the real world, in virtual reality or in a third-party 3D application.

5. Conclusions

In this paper we reported an accurate “Learning from Demonstration” robot programming method. By means of a series of sensors, the robot programmer’s movements are accurately captured and then transmitted to the robot, which is then able to exactly reproduce the learned movements.

We also introduced current work on the development of a virtual- and an augmented-reality environments that facilitate cobots workspace design and robot real-time operation, in view of a maximum-safety collaboration. Our system is still open for further developments, and we are adding new features on a periodic basis. For the real-time robot operation, the goal of our approach is not to replace traditional security measures such as safety switches and lasers. The purpose is rather to add an effective security layer for maximum operator safety and to reduce the need to interrupt the production.

The developed system also includes the following features: 1) Hazardous areas are shown directly on the scene, as can be seen in Fig. 9. This should reduce the risks of the operator walking into a protected area. 2) A *safety score*, which is based on the combination of multiple inputs, such as: a) the operator’s distance to the robot, b) the safety level of the current robot’s task, c) the operator’s attention (e.g., tracking the operator’s eyes) relative to the robot, and d) the operator’s (next/imminent) movement prediction. The value of the *safety score* is used to warn the operator by using audio and visuals cues (when the score is too low). 3) The system can also be equipped with a signal to communicate with the robot to slow it down or stop it when the operator gets too close. We consider that our solution may lead to increased productivity, as it provides a common User Interface (UI) directly in front of the robot, which allows the operator to manage tasks and eventually stop the robot. Additionally, when the robot is stopped, the operator can jog the robot from a safe distance, without touching it, simply by moving 3D spheres around in augmented reality.

References

[1]. A. Rizzotti-Kaddouri, M. Kunze, L. Jeanneret, L. Depierraz, Collaborative Robots and Set of Sensors for Learning by Demonstration, in *Proceedings of the 3rd ISFA Winter Conference on Automation, Robotics, & Communications for Industry 4.0/5.0 (ARCI 2023)* Chamonix-Mont Blanc, France, 22-24 February 2023, pp. 207-209.

[2]. J. Lee, A survey of robot learning from demonstrations for human-robot collaboration, *arXiv preprint arXiv:1710.08789*, 2017.

[3]. Z. Zhu, H. Hu, Robot learning from demonstration in robotic assembly: A survey, *Robotics,* 2018, Vol. 7, No. 2, pp. 17-42.

[4]. J-T. Masse, Capture de mouvements humains par capteurs RGB-D, Thèse de doctorat, *Université Paul Sabatier-Toulouse III,* 2015.

[5]. K. Bernardin, K. Ogawara, K. Ikeuchi, A sensor fusion approach for recognizing continuous human grasping

sequences using hidden Markov models, *IEEE Transactions on Robotics*, 2005, Vol. 21, No. 1, pp. 47-57.

[6]. Y. Lan, J. Li, Z. Ju, Data fusion-based real-time hand gesture recognition with Kinect V2, in *Proceedings of the 9th IEEE International Conference on Human System Interactions (HIS' 2016)*, 2016, pp. 307-310.

[7]. Z. Zhu, H. Hu, D. Gu, Robot performing peg-in-hole operations by learning from human demonstration, in *Proceedings of the 10th Computer Science and Electronic Engineering (CEEC' 2018)*, 2018, pp. 30-35.

[8]. M. Koskinopoulou, S. Piperakis, P. Trahanias, Learning from demonstration facilitates human-robot collaborative task execution, in *Proceedings of the 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI' 2016)*, 2016, pp. 59-66.

[9]. H. Ravichandar et al., Recent advances in robot learning from demonstration, *Annual review of control, robotics, and autonomous systems*, 3, 2020, pp. 297-330.

[10]. S. E. Ovur, et al., Novel adaptive sensor fusion methodology for hand pose estimation with multileap motion, *IEEE Transactions on Instrumentation and Measurement*, 70, 2021, pp. 1-8.

[11]. I. Soares, et al., Accuracy and repeatability tests on HoloLens 2 and HTC Vive, *Multimodal Technologies and Interaction*, Vol. 5, No. 8, 2021, pp. 47-62.

[12]. Automatic planning and scheduling: https://en.wikipedia.org/wiki/Automated_planning_and_scheduling#Classical_planning

[13]. STRIPS: https://en.wikipedia.org/wiki/Stanford_Research_Institute_Problem_Solver

[14]. M. Faroni et al., A layered control approach to human-aware task and motion planning for human-robot collaboration, in *Proceedings of the 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN' 2020)*, 2020, pp. 1204-1210.

[15]. HTC. Vive website. Vive Pro Eye. https://www.vive.com/

[16]. Unity Technologies, https://unity.com

[17]. HTC. Vive website, Vive Tracker sensor, https://www.vive.com/

[18]. D. Xu, S. Nair, Y. Zhu, Neural task programming: Learning to generalize across hierarchical tasks, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA' 2018), 2018,* pp. 3795-3802.

[19]. M. Stenmark, E. A. Topp, From demonstrations to skills for high-level programming of industrial robots, in *Proceedings of the 2016 AAAI Fall Symposium*, 2016, pp. 75-78.

[20]. Microsoft HoloLens 2. https://en.wikipedia.org/wiki/HoloLens_2