SIGraDi2020
Transformative design

XXIV International Conference
of the Iberoamerican Society
of Digital Graphics
Medellín | Colombia

Robotic Apprentices: Leveraging Augmented Reality for Robot Training in Manufacturing Automation

Eduardo Costa
The Pennsylvania State University | United States of America | castroecosta@gmail.com

José Duarte
The Pennsylvania State University | United States of America | jxp400@psu.edu

Sven G. Bilén
The Pennsylvania State University | United States of America | sbilen@psu.edu

Abstract

In the scope of Industry 4.0, a framework is proposed to leverage the potential of articulating Augmented Reality and Robotic Manufacturing in the construction industry. The objective of such framework is to enable robots to learn how to perform tasks using direct interaction with human operators. As a first step, we established a connection between a robot and its trainer—or controller—in which the robot mirrors the operator's actions. Augmented Reality hardware is used for capturing the trainer's gestures and the surrounding environment. A digital tool was implemented using Grasshopper and additional plugins to control the process.

Keywords: Augmented reality; Robotic arm; Programming by demonstration; Human–Robot Collaboration; Industry 4.0.

INTRODUCTION

Applications of digital technologies such as augmented reality (AR), machine learning (ML), and robotic manufacturing (RM) have been developing rapidly in recent years due to increased availability and reduction in cost. These technologies are widely considered to be enablers of the next industrial revolution, what is referred to as Industry 4.0 (McKinsey Digital, 2015). AR can potentially improve operator productivity, whereas ML enables a better use of production data generated in industrial processes. Finally, RM has the potential of both enhancing labor capacity and reducing labor costs and safety hazards in industrial manufacturing.

Despite the decreasing costs of RM systems, the complexity of programming and operating a robotic arm can still constitute an obstacle for integrating such systems in industries such as architecture, engineering, and construction (AEC), which have traditionally taken longer to adopt technological innovation than some other industries, for example, the automotive industry (Gann, 1996). Another characteristic of such industries is that they deal with customized manufacturing scenarios, requiring more flexibility than their mass-production counterparts. For these scenarios, an RM system needs to learn more than just how to repeat a sequence of tasks.

Typically, industrial robots are programmed by technicians or engineers whose expertise is programming robots, rather than performing the tasks for which the robot is being programmed. However, by "learning" from an experienced manufacturing line operator, the robot is likely to perform a manufacturing task better than by being "trained" by a robot expert. Therefore, we propose exploring a system in which a robot can learn particular tasks by mimicking the actions of a manufacturing line operator, similarly to how an apprentice learns from an expert.

In the role of expert, the operator performs their normal tasks while wearing an AR headset (ARH), which enables capturing data related to the actions needed for completing those tasks. The captured data would comprise a geometrical mesh corresponding to the workspace and the objects in it, and the orientations of the operator's hands, both of which can be captured by currently available commercial ARH solutions.

In an initial phase, the captured data can be used, either after the fact or in real-time, to directly instruct the robot—the apprentice—on how to perform the task by repeating the operator's actions. In a subsequent phase, the data captured by ARH can be used to train the robot through ML, enabling it to perform the task in contexts that have not been captured during the training. A possible case study to support this research is an ongoing project that focuses on additive manufacturing of concrete structures.

RELATED WORK

The concept of teaching robots to perform tasks by example, known in the field of robotics as "programming by demonstration" (PbD) or "imitation learning", has been explored for the last three decades, and is considered a powerful alternative to robot preprogramming due to three factors: it reduces complexity of search spaces for learning, it offers a natural means of interacting with robots, and it helps understand the coupling of perception and action (Billard et al., 2008). While the concept of teaching a robot by example is not novel, current availability and, consequently, accessibility to AR and RM technologies have reignited the interest of researchers in the fields of construction and manufacturing in this approach, spawning a number of projects relatable to our own.

Pérez et al. (2019) explored a framework for training robot operators with an emphasis on safety by employing Virtual

Reality (VR) to simulate a digital twin of a real robotic arm. Their framework is oriented toward cost-effectiveness by making use of low-cost digital tools such as Blender for modelling and rendering a digital twin of the environment surrounding the real robot, and Unity3D for implementing the user interface. Despite implementing teleoperation capabilities, this approach is oriented towards training robot operators rather than training robots themselves.

Betti et al. (2020) articulated AR and RM to enable non-expert users to design, produce, and assemble a brick tower. An ARH is primarily used for designing the tower, whereas a robotic arm attached with a hot-wire-cutter end effector subsequently fabricates the bricks. Finally, the ARH is used to show the user how to manually assemble the tower. The project relies on a custom-developed multi-modal interface, which includes a button-based graphical user interface (GUI) to guide the process.

Looking beyond a construction context, Yew et al. (2017) implemented a teleoperation system for maintenance robots using an AR environment as a human–robot interface, by mapping the actions of a virtual robot visualized in the ARH and a real robot at a remote site. This project relies on the use of fiducial markers to map between environments surrounding both the real and the virtual robot, which implies a previous knowledge and preparation of both these environments. In fact, one of the suggested paths for future improvement is enabling the system to detect and avoid unknown objects surrounding the real robot in the remote site.

Finally, Puljiz & Hein (2019) explore the use of AR for facilitating a number of tasks in the context of human–robot collaboration (HRC), such as during the set-up, programming, and interaction phases. A relevant aspect of this project is its use of a number of functionalities existing in current ARHs other than merging virtual and physical environments, such as determining its location within a physical environment; producing a digital geometric model of the environment, and tracking hand location and gestures, similarly to the project presented in this paper.

METHODOLOGY

In this paper we present the first step toward the implementation of Robotic Apprentices, which consists of developing a framework that established a link between a human trainer and a robotic arm. Our approach towards materializing such linkage was having the robot mirror the physical movements of the human trainer.

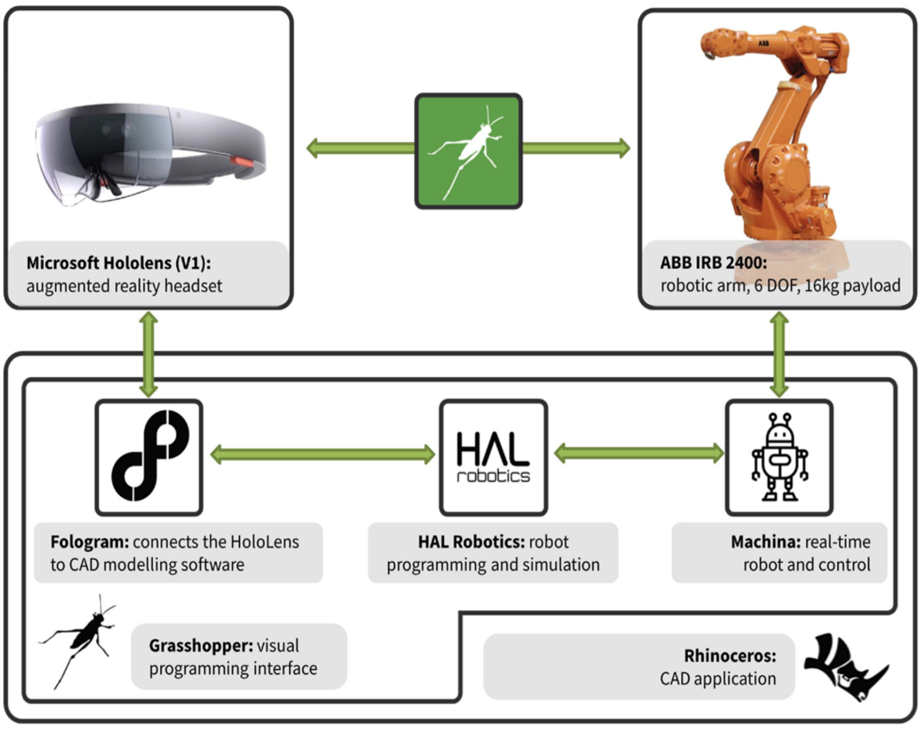
Microsoft Hololens (V1): augmented reality headset
ABB IRB 2400: robotic arm, 6 DOF, 16kg payload
HAL robotics
Fologram: connects the HoloLens to CAD modelling software
HAL Robotics: robot programming and simulation
Machina: real-time robot and control
Grasshopper: visual programming interface
Rhinoceros: CAD application

Figure 1: System architecture for Robotic Apprentices

Given the capabilities of current ARHs in capturing a trainer's position within an environment and the orientation and gestures of their hands, a connection was implemented between an ARH and a robotic arm. Our present setup articulates a Microsoft Hololens ARH version 1 (Microsoft, 2016) and an ABB robotic arm model IRB 2400 (ABB, 2008) with 6 degrees of freedom and a maximum payload of 16 kg. For prototyping the connection between these two elements we used Grasshopper (Rutten, 2009), a visual programming interface for Rhinoceros (Robert McNeel & Associates, 1993), a CAD application. These two pieces of software are popular in the AEC industry given their lower cost and fast learning curve. Additionally, three plugins for Grasshopper (GH) are used: Fologram, HAL Robotics, and Machina.NET (Figure 1).

Fologram (2018) synchronizes geometry from Rhinoceros or Grasshopper with geometry rendered on mixed reality devices such as the Hololens over a local WiFi network connection. A relevant application of such tool is a mixed reality platform that enables construction workers to carry out tasks by following interactive holographic instruction sets generated directly from parametric models, aiming at reducing construction time and improving training, collaboration and skill development (Jahn et al., 2020). In our prototype, Fologram is used to establish a connection between Hololens worn by the user and the GH model that controls the robotic arm.

HAL Robotics (2012) is a GH plugin that aims to improve industrial robots programming ergonomics, in order to facilitate their use in architecture teaching and research contexts, making it compatible with simulation and instruction generation methods used for robot programming and control (Schwartz, 2013). In our prototype, we used specific HAL functionalities to simulate and control the robotic arm, namely inverse kinematics and collision detection. Since the standard version of HAL does not allow for real time control of the robotic arm, an additional plugin was needed.

Machina.NET (García del Castillo y López, 2016, 2019) is a library for programming and control of industrial robots, designed to build applications that interface with robotic devices in real time. Providing an immediate and intuitive entry point to real-time robot control, Machina is particularly suitable for controlling systems that require concurrent responsiveness to sensory or user input. In out prototype, this library is used within a GH plugin to enable real-time control of the robotic arm through GH.

DEVELOPMENT

The prototype framework was set up in a laboratory equipped with the aforementioned ABB robot. In addition to the robot and the Hololens ARH, the framework's hardware comprises an ABB IRC5 controller for the robot and a VR-ready Alienware 15 laptop for running the necessary software.

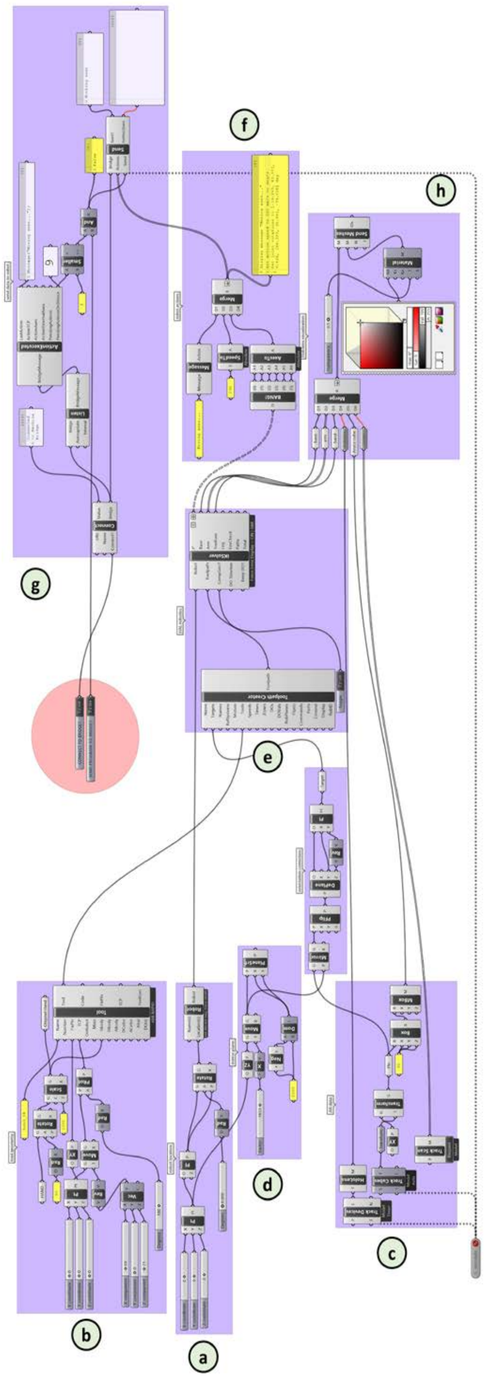

Figure 2: Grasshopper definition connecting the Augmented Reality Headset to the Robotic Arm

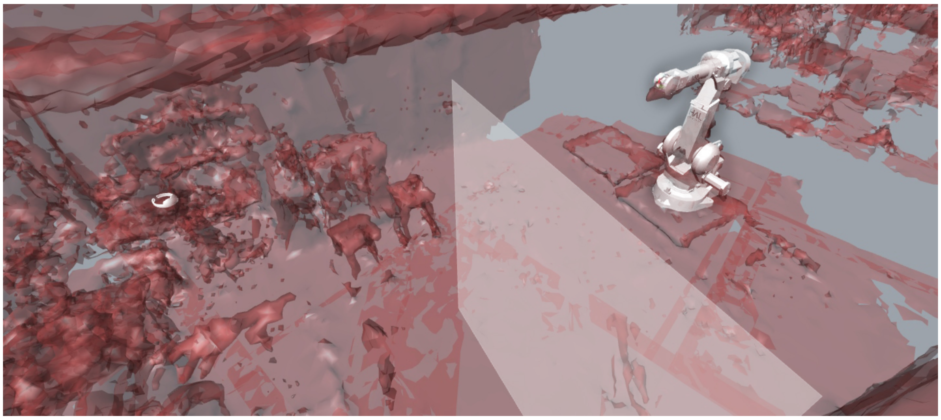

Figure 3: Geometry of the working environment, captured in real-time by the ARH (in red). Representations of the ARH and the Robot (in white), and representation of the mirror plane.

In our current setup, the robot mimics the movements of the trainer across a vertical mirroring plane, which coincides with the safety laser curtain that ensures safety while operating the robot. For controlling the robot, the trainer must be wearing the ARH so it can perform its main function: to capture the trainer's movements and the surrounding environment. In typical uses of AR in construction and industrial settings, the user is presented with a GUI for performing a particular task, which in our case is controlling the movements of a robotic arm. In this project, however, the use of AR affords an alternative approach, i.e., graphical information is occasionally overlaid in the ARH. For example, during the development phase, the system is calibrated by visually matching the projection of the virtual robot onto its existing, physical counterpart. However, the main functionalities used from the ARH are detection of hand position and gestures, and 3D scanning of the surrounding environment.

SOFTWARE

The Robotic Apprentices framework is controlled by a GH definition that comprises eight groups of components, each responsible for the following tasks and supported by a specific plugin (indicated in parenthesis) (Figure 2): a) positioning the virtual robot (HAL Robotics), b) defining the end effector (HAL Robotics), c) acquiring ARH data (Fologram), d) calculating geometric data for the robot to follow (GH only), e) calculating joint angles through Inverse Kinematics (HAL Robotics), f) compiling robot actions (Machina.NET), g) sending data to the robot controller in real time (Machina.NET), and h) sending visualization data back to the headset (Fologram).

ROOM SETUP

At the beginning of each session, and as part of the default procedure for using Fologram, the system needs to be calibrated by defining a reference plane of the virtual space. In the proposed system, the origin of such plane matches the center of the robot's base. Currently, the calibration process is carried out by the trainer wearing the ARH, and since the virtual model of the robot is displayed, the calibration process is facilitated.

Additional visual cues can be used for the calibration, since the ARH is constantly capturing spatial data from its surrounding environment into a point cloud. Therefore, a geometric model of the room, which is continuously being updated, can be optionally displayed in the ARH as a mesh (Figure 3).

HAND TRACKING

Even though Hololens is able to track the trainer's hands without any fiducial markers, such tracking is limited to position only. In order to determine additional geometric information such as orientation, an Aruco cube is attached to the trainer's hand (Figure 4), which provides more precise positional tracking. In the present setup, the trainer's hands and the attached Aruco cubes must be within the trainer's field of vision in order to be tracked. Reportedly, improved hand-tracking capabilities announced for the next generation Hololens 2 should remove the need for such markers.

END EFFECTOR

At this stage, our objective was to ensure that the robot accurately mirrored the trainer's hand movements, both in terms of position and orientation. Therefore, for testing purposes, a 3D-printed artificial hand was attached to the physical robotic arm as an end effector, whereas the 3D-model that was used for printing was attached to the virtual robot (Figure 4). Such approach allowed for quickly and visually assessing if the robot's last joint was correctly oriented.

DISCUSSION

Running the GH definition, we can see the robot in action, mimicking the operator to the best of its abilities. Work sessions were captured in video by the ARH, from which a number of observations were possible.

One of the first observations—and a rather unexpected one—is that the movement of the physical robot is often temporally ahead of the virtual robot overlaid in the ARH. Although at first unexpected, this time lag can be explained by a differential in information flow. In fact, the GH definition calculates the robot's pose and transmits it along two streams with different types of information: the first stream consists of Machina.NET actions corresponding to joint rotations calculated by IK, whereas the second stream contains the geometric data representing the robot itself, and sends them wirelessly to the ARH to be rendered one frame at a time. The difference in the amount of data, and the difference in transmission media (wired vs. wireless) should be responsible for the delay between the real and the virtual movements.

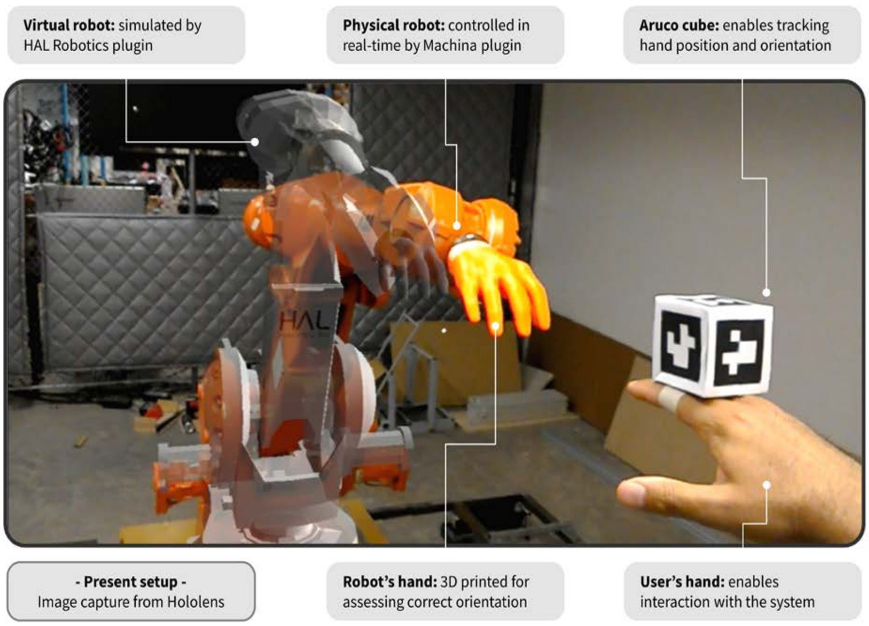
Virtual robot: simulated by HAL Robotics plugin
Physical robot: controlled in real-time by Machina plugin
Aruco cube: enables tracking hand position and orientation
- **Present setup** -
Image capture from Hololens
Robot's hand: 3D printed for assessing correct orientation
User's hand: enables interaction with the system

Figure 4: Using Robotic Apprentice (image captured from Hololens)

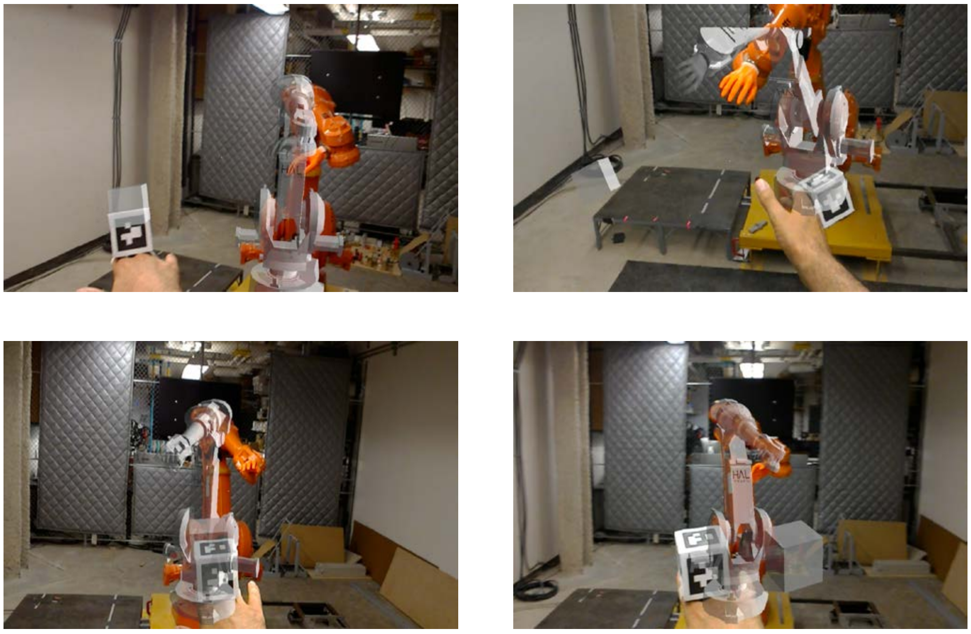

Figure 5: Robotic arm mimicking four different positions of the user's arm (image captured from Hololens)

Another observation is that the system works well from the standpoint of safety, both for the operator and for the robot. The safety laser curtain effectively prevents the trainer from getting too close to the robot and shuts the system down if it is breached. On the other hand, the IK module prevents the robot from trying to reach positions that are out of range, avoiding too much strain on the robot and subsequent malfunction.

A final observation is that the robot occasionally performs multiple full rotations of the wrist joint. Besides being unnecessary, such excess rotations impose additional wear on the robot, and should be fixed. This occurrence is likely to happen when the robot moves too close past a singularity point in the robot trajectory space and should be researched further. To fix this issue, it might be necessary to optimize the IK module or to remap the trajectory space.

Despite the issues mentioned, the Robotic Apprentices framework was relatively straightforward to assemble. The cost of assembling it was relatively accessible: all the software used in this project is publicly available and some of it is free to use. Moreover, bear in mind that this was a prototype framework, and that further development towards an integrated application should render some of the plugins unnecessary.

CONCLUSION

This is only the first step towards enabling the robotic arm to replicate the actions of its operator. Future research work includes integrating functional end-effectors such as grippers, and using actual construction materials, such as bricks or window frames.

The system's capabilities can also be extended by integrating robot vision components. In that case, the robot's field of view could be transmitted back to the ARH, providing the trainer with more precise control over the robot's actions.

In a final phase of this project, we intend to enable the system to learn with the actions that it replicates using ML, towards true Robotic Apprenticeship.

ACKNOWLEDGEMENTS

The research presented in this paper is funded by the Industry 4.0 Workshop by The Learning Factory, and results from a collaboration between the Stuckeman Center for Design Computing (SCDC) and the School of Engineering Design Technology and Professional Programs (SEDTAPP) at the Pennsylvania State University. Support provided by the Harold & Inge Marcus Innovation & E-Ship Fund.

REFERENCES

ABB. (2008). *IRB2400*. https://new.abb.com/products/robotics/industrial-robots/irb-2400

Betti, G., Aziz, S., & Ron, G. (2020). Pop-Up Factory: Mixed Reality Installation for the MakeCity Festival 2018 in Berlin. In *Impact: Design With All Senses* (pp. 265–276). Springer International Publishing. https://doi.org/10.1007/978-3-030-29829-6_21

Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2008). Robot Programming by Demonstration. In *Springer Handbook of Robotics* (pp. 1371–1394). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-30301-5_60

Fologram. (2018). *Fologram*. https://fologram.com/

Gann, D. M. (1996). Construction as a manufacturing process? Similarities and differences between industrialized housing and car production in Japan. *Construction Management and Economics*, *14*(5), 437–450. https://doi.org/10.1080/014461996373304

García del Castillo y López, J. L. (2016). *Machina.NET*. https://github.com/RobotExMachina/Machina.NET

García del Castillo y López, J. L. (2019). Machina.NET: A Library for Programming and Real-Time Control of Industrial Robots. *Journal of Open Research Software*, *7*(1). https://doi.org/10.5334/jors.247

HAL Robotics. (2012). *HAL Robotics*. https://hal-robotics.com/

Jahn, G., Newnham, C., van den Berg, N., Iraheta, M., & Wells, J. (2020). Holographic Construction. In *Impact: Design With All Senses* (pp. 314–324). Springer International Publishing. https://doi.org/10.1007/978-3-030-29829-6_25

McKinsey Digital. (2015). *Industry 4.0: How to navigate digitization of the manufacturing sector*.

Microsoft. (2016). *Hololens*. https://docs.microsoft.com/en-us/hololens/hololens1-hardware

Pérez, L., Diez, E., Usamentiaga, R., & García, D. F. (2019). Industrial robot control and operator training using virtual reality interfaces. *Computers in Industry*, *109*, 114–120. https://doi.org/10.1016/j.compind.2019.05.001

Puljiz, D., & Hein, B. (2019). *Concepts for End-to-end Augmented Reality based Human-Robot Interaction Systems*. http://arxiv.org/abs/1910.04494

Robert McNeel & Associates. (1993). *Rhinoceros*. https://www.rhino3d.com/

Rutten, D. (2009). *Grasshopper*. https://www.grasshopper3d.com/

Schwartz, T. (2013). HAL: Extension of a visual programming language to support teaching and research on robotics applied to construction. In *Rob | Arch 2012* (pp. 92–101). Springer Vienna. https://doi.org/10.1007/978-3-7091-1465-0_8

Yew, A. W. W., Ong, S. K., & Nee, A. Y. C. (2017). Immersive Augmented Reality Environment for the Teleoperation of Maintenance Robots. *Procedia CIRP*, *61*, 305–310. https://doi.org/10.1016/j.procir.2016.11.183