Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 76 (2018) 155–160

www.elsevier.com/locate/procedia

7th CIRP Conference on Assembly Technologies and Systems

Intuitive Robot Programming Using Augmented Reality

Sebastian Blankemeyer[a,*], Rolf Wiemann[b], Lukas Posniak[c], Christoph Pregizer[c], Annika Raatz[a]

[a]*Leibniz Universität Hannover, Institute of Assembly Technology, An der Universität 2, 30823 Garbsen, Germany*
[b]*Leibniz Universität Hannover, student*
[c]*Viewlicity GmbH, Schäferkampsallee 42, 20357 Hamburg, Germany*

[*] Corresponding author. Tel.: +49-511-762-18249; fax: +49-511-762-18251. *E-mail address:* blankemeyer@match.uni-hannover.de

---

Abstract

The demands on companies caused by the markets are becoming more and more fast-moving and complex. Reasons are the increasing number of variants of products as well as reduced product life cycle times. This is particularly relevant for assembly tasks, since a very high flexibility and adaptability to varying ambient condition must be ensured in this area. Therefore, a lot of steps are currently being carried out manually. However, if an automation is attempted in the field of assembly, companies are often facing a trade-off between a high degree of automation and flexibility of the production system. A concept to increase the automation rate in assembly tasks can be seen in the use of collaborative robots, so that the benefits of both, humans and robots, can be combined to accomplish the task. In doing so, one key issue is to simplify and thus to accelerate the programming process so that the necessary programming skills of employees can be reduced. Although some of today's collaborative robots already offer good programming approaches like kinesthetic teaching, this article introduces a new and more intuitive programming method which is based on Augmented Reality. For this purpose, components of an assembly group are virtually linked with CAD models by using optical markers. The operator can then virtually assemble the components according to the assembly sequence. Results of first tests indicate, that once the assembly process is recorded, the robot can accomplish the assembly in reality. Finally, possibilities for future developments are presented.

Keywords: augmented reality, intuitive robot programming, human-machine-interaction

---

1. Introduction

Short product life cycles and a high variety of designs demand flexible and intuitive programming methods for industrial robots so that companies can efficiently react to rapidly changing market situations. Conventional programming methods, such as teach-in methods or simulation programs, require specialized knowledge so that fast and flexible changes are difficult to realize [1,2]. For this reason, different approaches can be found in literature which aims at abstracting the programming procedure so that the robot's program code is automatically generated [3]. Intuitive programming methods are developed especially in the field of human-robot collaboration. Collaborative robots enable humans and robots to work together within the same workspace and are already used in industry. These robots can be guided by hand, whereby points or trajectories can be stored (kinesthic teaching). Thus, collaborative robots offer a suitable hardware design for flexible and simple programming. On the other hand, the software is often simplified by using program blocks to abstract the textual programming [4,5].

A further simplification of programming can be achieved by translating human actions directly into a robot program. Programming-by-demonstration (PbD) procedures are based on the imitation of human actions by the robot. By deriving motion paradigms, the demonstrated actions can also be transferred to other or similar tasks [6].

Another approach of programming robots is the use of gestures. In order to generate waypoints for determining the path of the robot, camera systems and special software tools are used to recognize such gestures and interpret them, respectively [7]. If such a camera system is combined with an augmented reality system, additional information and objects can be visualized spatially. Previous research projects on the use of augmented reality (AR) in the field of robot programming have used AR systems, which are, however, complicated and usually not available [8,9]. The development of powerful head-mounted displays (HMD) such as the Microsoft Hololens enables a highly simplified system structure. The Microsoft Hololens combines the AR and the camera system in a compact device. A permanent tracking of the environment makes it possible to place 3D-holograms into it. Furthermore, the integrated gesture and speech recognition permit an intuitive control and interaction with holograms. Since these functionalities are beneficial for the design of intuitive robot programming methods, the Hololens is used as an interface in this work. First, the real

2212-8271 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 7th CIRP Conference on Assembly Technologies and Systems.
10.1016/j.procir.2018.02.028

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

components are recognized by markers and overlaid with a virtual model. This can be displayed by the hololens directly at the workstation. The worker can then virtually assemble the product using gestures. The gestures are recognized by the integrated cameras and the assembly steps are thus recorded by the Hololens. Finally, the robot can carry out the process in reality through the acquired data.

At the beginning of this paper, an overview on already existing AR systems is given, which have been used and developed in the field of human-robot-collaboration. The concept and the implementation of a new programming method are then described, which will also be examined with regard to its accuracy. Based on a summary, an outlook on future work is given.

2. Augmented Reality in Robotics

Augmented reality can be described as a computer-aided transfer of virtual aspects to the real world. The Hololens is a device that can be used for AR applications. A characteristic of an AR system is the combination of the reality with a virtual world, an interaction in real time and a three-dimensional registration [10]. Based on these characteristics, an AR system usually consists of a display for visualizing virtual contents, a camera system for tracking the environment and communication interfaces. Basically, a distinction can be made between head-mounted displays (HMDs), handheld devices and projection systems, which have often intuitive control interfaces such as gesture and voice control.

Despite the development of more powerful hardware and software, the usage of AR systems is mainly limited to research and gaming applications. Main applications comprise installation and maintenance tasks [11,12]. In industrial practice, for example, Boeing successfully uses Google Glass to visualize assembly instructions [13]. M. Funk [14] developed various systems, with which employees are supported in assembling and commissioning tasks by virtual information and instructions, that are visualized via a projector. In addition, the user's actions are detected with a camera. O. Sand et al. [15] developed a similar system to provide assembly and picking instructions, which is based on a projector as well.

The possibility to display information about the assembly process is also used in the field of human-robot collaboration. S. Makris et al. [16] developed a system, with which assembly steps can be visualized and information about the sequence of events can be displayed. In addition, the workspace and the planned trajectory as well as the current status of the robot can be visualized. An AR glass is used to display the virtual content and the system can be controlled via a smartwatch. O. Danielsson et al. [17] developed a similar system, which supports the worker with virtual instructions during the collaborative assembly. In doing so, the user receives information about the assembly steps performed by or in cooperation with the robot or about assembly steps of the user himself. The AR contents are visualized using a display and the components are detected by means of a marker tracking procedure.

Furthermore, there are systems that not only display information and instructions, but also support the user while programming a robot. In this case, the use of AR systems is focused on path planning and the simulation of the created robot program in order to detect and avoid collisions. H. Fang et al. [18] developed a method to show points in space by a hand-guided marker. Based on this, a path is calculated which can be simulated with a virtual robot model and transferred to the real robot. A laptop is used to display the virtual content. J. Chong et al. [19] developed a similar system, where besides saving points, it is also possible to show a trajectory. The procedure presented by Y. Pai et al. [20] is based on the same functional principle but it has been extended to include the possibility of programming pick-and-place tasks. Virtual information and content is visualized via a conventional computer monitor. J. Lambrecht [8] uses gestures that are detected by the Microsoft Kinect instead of markers. With the created system, real objects can be overlaid with virtual objects. In addition to path planning and simulations, simple pick-and-place tasks can also be programmed by manipulating the virtual parts and translating this action into a robot program. A tablet computer is used to control the system and to show the AR content. G. Reinhart [9] developed a system for programming trajectories using a pen. This is particularly suitable for programming in the field of welding and laser hardening. In comparison to the conventional teach-in method, a considerably time saving approach could be shown.

When considering the presented systems in the field of robot programming, it is noticeable that they are characterized by an intricate structure, which generally consists of a camera system and an AR display. In addition, tablets or laptops are often used to display AR contents, however, such systems are impractical in application and do not allow for free-handed operations. In contrast, the Microsoft Hololens connects a camera display with an AR display inside an HMD, which yields a simple system setup in total. In addition, unlike many other HMDs, holograms can be displayed in 3D and a permanently scanning process of the environment allows the user to place and fix holograms in the workspace. Furthermore, the gesture and voice control provides an intuitive control functionality.

3. Concept of programming

The goal of this work is to develop an intuitive programming tool for robots. As shown in the previous section, AR is a method to allow human-machines interactions. In this context, the Hololens offers new possibilities to realize communication and information processing, which are used to develop an application that converts virtual assembly steps into a robot program. In this way, the first step will be the generation of pick-and-place motion profiles for the robot to assemble components (see Fig. 1).

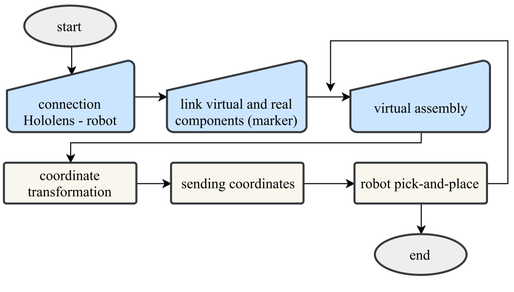

Fig. 1. Flowchart of the developed robot programming system

The process design is structured in such a way that the worker wears the Hololens which is linked with the robot at his workplace. The components to be assembled can be reached by the robot and lie on optical markers that are needed for localization. Furthermore, the markers allow the connection between real and virtual components. They are necessary because the internal tracking possibilities of the hololens only allow the creation of a spatial mapping, but not the recognition of objects. Therefore, the *Software Development Kit* SDK developed by Vuforia is used for the hololens to determine the position and orientation of the components.

In order to program an assembly step, the user first of all carries out the assembly step virtually by moving the virtual component to the desired position. The virtual component is displayed to the user by the Hololens and is at the same position as the real component, whose position is determined by marker tracking. The start and end coordinates of the assembly step are determined and saved with respect to the internal coordinate system of the Hololens (CSH). To enable the robot to manipulate the real components, these coordinates must be transformed from CSH into the robot's base coordinate system (CSR). The CSR is already shifted to the mounting aid as it simplifies the transformation calculation (see Fig. 2). The transformed coordinates are subsequently sent to the robot, which then performs the manipulation in reality.

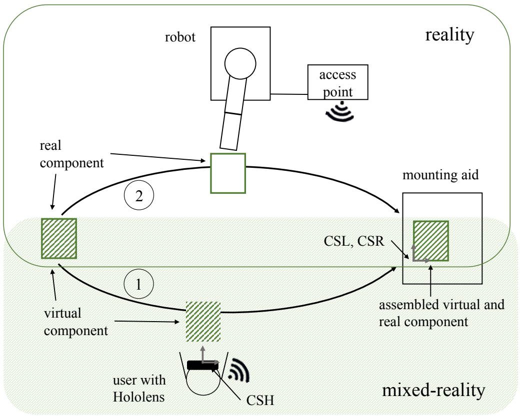

Fig. 2. Programming through virtual execution

In order to work with the obtained data, a program logic is implemented on the robot controller that uses the start and end coordinates of the moved component to automatically generate the assembly process. The path planning task between the individual poses is thus carried out directly by the robot controller.

The interaction with the virtual objects and the control of the application is realized via the Hololens' standardized communication interfaces. By focusing on individual objects, the user can select components and interact with them using the so-called *AirTap*, which is similar to a mouse click. Based on this, the virtual components can be controlled and positioned with the help of manipulation gestures.

4. Implementation

Windows 10 is used as the operating system for Microsoft Hololens. The *Universal Windows Platform* (UWP) serves as a uniform development platform for Windows 10 applications. For the development of Hololens applications, Visual Studio is used in combination with the game development software Unity3D, for which Microsoft already provides basic functions such as gesture and speech recognition.

The basis of the application is the product, which is to be assembled virtually by the user as well as in reality by the robot. To display the product holographically, the CAD model of the product has to be converted to a mesh model and then imported into Unity3D. During this process, the assembly structure defined in the CAD model is retained. However, it can be changed in Unity3D as required. The program logic is created by scripts which are designed in such a way that they can be used on any model and thus facilitates the exchange of different models.

A toolbar with buttons is implemented as user interface. So, all functions of the programm can be called up and settings can be made. In addition, a voice control allows an intuitive control of the application and the user is given feedback and instructions via a voice output.

4.1. Use case

To demonstrate the functionality by means of a sample application, this paper is concerned with the implementation of the assembly process of a helicopter model. The helicopter serves as a use case in the learning factory of the Institute of Production Systems and Logistics. For simplification, only two components are considered, but the application can be extended by additional components without any problems. Two skirting boards of the helicopter model are used as components, which are assembled on a mounting aid (see Fig. 3). In this work a Kuka KR6 R900 sixx robot is equipped with a suction gripper and connected to the Hololens via the Kuka technology package *Robot-Sensor-Interface* (RSI). The RSI enables real-time data exchange between the Hololens and the robot, based on the *User Datagram Protocol* (UDP).

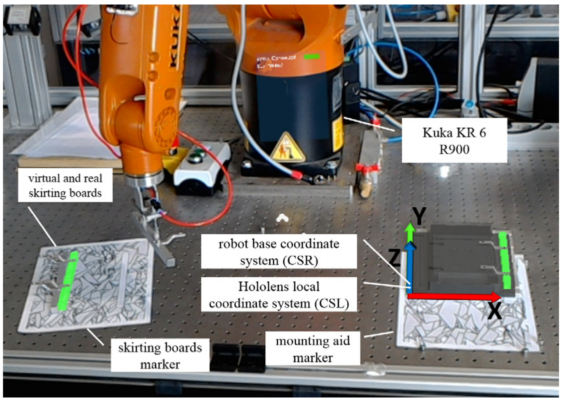

Fig. 3. Assembly cell with real and virtual assemblies

4.2. Marker-Tracking

In order to be able to overlay and link the real components with the respective virtual components, the real components must be detected in the environment. Furthermore, a fixed spatial reference point has to be defined in order to transform poses from the Hololens' coordinate system (CSH) into that of the robot (CSR) and vice versa. The Vuforia SDK available for the Hololens is used for the marker tracking.

The markers themselves are generated using the tool of AR-software developer Brosvision [21]. The tool takes all specifications into account, which are recommended by Vuforia to

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

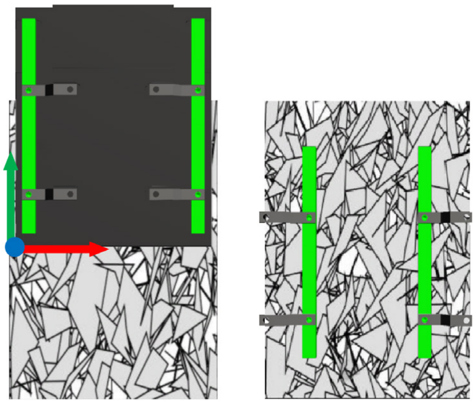

Fig. 4. Marker with mounting aid and Hololens local coordinate system (left) and virtual components (right)

get the most accurate and reliable tracking results. The markers are printed in DIN A4 size and placed within the robots' workspace. There is a defined position for the real components on each marker. In Unity 3D, this position on the marker is also assigned to the virtual object, so that the real object is aligned with the virtual object (see Fig. 4).

As soon as a marker has been detected, the virtual component is shown and thus overlays the real component (see Fig. 3). In this way, all markers are sequently scanned. The tracking algorithm will be switched off to save computing resources if all markers have been detected. However, it can be switched on again at any time to rescan the markers.

In addition, the markers are used to reference the robot's coordinate system CSR with respect to the CSH. In doing so, coordinates can exchanged. Therefore, this marker must be fixed within the robot workspace. The other marker can be positioned arbitrarily.

4.3. *Virtual assembly*

For a virtual assembly of the model, interaction and manipulation of each component must be possible. As shown in Fig. 3, the complete product is assembled on the marker with the mounting aid. The target position of each subassembly is defined by the assembly model, which is placed on the mounting aid but not visualized.

The components (here: skirting boards of the helicopter) are moved with a manipulative gesture. This gesture is predefined by Microsoft and is therefore straightforward to implement. To handle an object virtually, the user focuses a component and remains the *AirTap* as long as the movement is in progress. The object can be moved simulatanously to the hand movement and placed in the environment. If the component is near the target position that can be determined from the assembly model, it changes its color to signal to the user that the object can be placed. The workpiece is then set to the target position when the manipulation gesture is terminated. Afterwads, the start and end coordinates of this assembly step are transformed and sent to the robot, which manipulates the real component.

4.4. *Coordinate transformation*

In order to be able to exchange the start and end coordinates of the assembly process between the Hololens and the robot, they must be transformed (see Fig. 5). As soon as the marker of the mounting aid has been detected, the local coordinate system of the Hololens (CSL) is superimposed on the robot's base coordinate system. Basically, poses of objects are specified by the Hololens in a global coordinate system. Therefore, the start and end coordinates are first transformed into the local coordinate system defined on the mounting aid marker. This is followed by a transformation into the robot's base coordinate system. Since the Hololens application is programmed with Unity3D, which uses a left-handed coordinate system, the coordinates are finally transformed into the robot's right-handed coordinate system. With the described transformation, only the position coordinates and not the rotation coordinates are transformed. With the developed application, components could only be assembled in z-direction so that only the rotation around the z-axis is calculated.

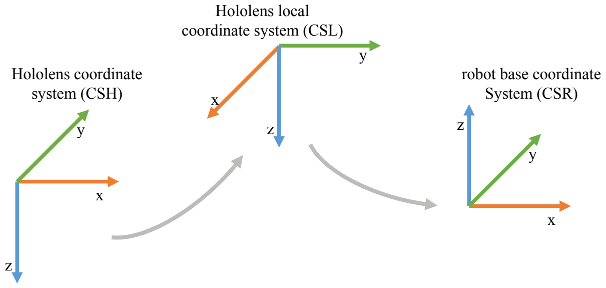

Fig. 5. Coordinate transformation between the Hololens and the robot

4.5. *Robot program and data exchange*

After the coordinates have been transformed, they are sent to the robot by the Hololens. The Kuka technology package *KUKA.RSI* is used for data exchange. With this, a real-time connection between the robot and an external system (here: Hololens) can be established. The Hololens is connected directly to the robot via a local network and so that a simple system setup is achieved. For data exchange and the program logic of the assembly process, a robot program is implemented in *KukaRobotLanguange* KRL, which is a special programming language for Kuka robots. The coordinates sent by the Hololens are put in the defined variables in this program. If the coordinates and a start signal are received from the Hololens, the programmed assembly sequence is started (see Fig. 6). Starting from a waiting position, the starting pose is approached with a PTP movement and at first with an adjustable offset in vertical

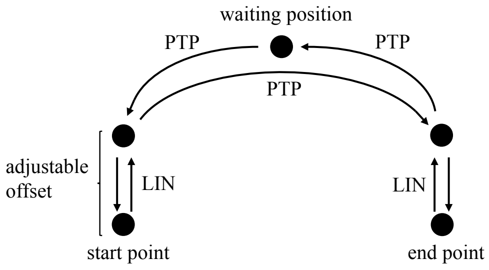

Fig. 6. Robot pick-and-place motion

direction. This is followed by a linear movement and the component is picked up. The target pose is also approached with a PTP movement and an offset. The component is then placed with a linear movement and the robot moves back to the waiting position. The user can then virtually demonstrate the next assembly step and start the robot again.

5. Accuracy and precision of marker tracking

An important feature when considering assembly processes is the required accuracy. If the production system does not provide the required accuracy, an assembly is not useful. The marker tracking is crucial for the accuracy of the robot assembly. The more precise the tracking, the more accurately the robot can move and place the components. In order to examine the accuracy of the tracking, two markers are placed at a distance of 0.5 m within the workspace (see Fig. 7). Both markers are scanned from different distances and angles with the Hololens. In doing so, the user is located in the middle of the markers and scans the markers by looking to the right and to the left. After both markers have been scanned, the distance of the reference marker to the local coordinate system defined on the marker (see Fig. 7) is compared to the real distance of the markers. This setup and procedure corresponds to the method developed for robot programming, in which the coordinates of the assemblies are also specified in the local coordinate system of the Hololens (CSL) and are then transformed into the robot's base coordinate system.

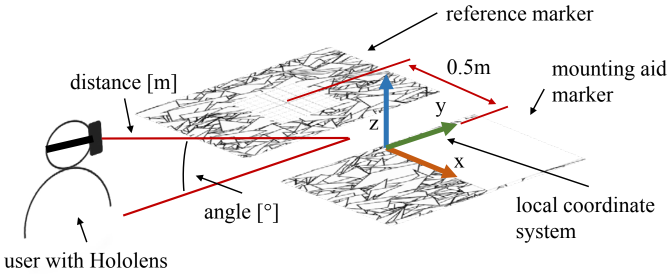

Fig. 7. Test setup to measure the accuracy of the Vuforia tracking

To examine the accuracy and precision of the tracking process, the distance and angle between the Hololens and the robot are varied. At each of the 25 examined measuring points (5 different angles and distances) 20 measurements are made. Moreover, ISO 9283 [22] is applied for the validation. ISO 9283 is mainly used in the field of industrial robots, but is also used for examinations of tracking systems [23]. The accuracy A_s is defined as follows:

$$A_s = \sqrt{(\bar{x}-x_{ref})^2 + (\bar{y}-y_{ref})^2 + (\bar{z}-z_{ref})^2}$$
$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i; \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i; \quad \bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i \tag{1}$$

The accuracy (A_s) indicates the center of gravity of the point cloud, i. e. the distance between the mean value of the measured values ($\bar{x}, \bar{y}, \bar{z}$) and the reference value ($x_{ref}, y_{ref}, z_{ref}$). According to ISO 9283, the precision P_s is defined by the distance of the measured values to the mean value ($\bar{l}$) and the standard deviation (σ_l) and is calculated as follows:

$$\begin{aligned} P_s &= \bar{l} + 3\sigma_l \\ \bar{l} &= \frac{1}{n}\sum_{i=1}^{n} l_i \\ l_i &= \sqrt{(x_i-\bar{x})^2 + (y_i-\bar{y})^2 + (z_i-\bar{z})^2} \\ \sigma_l &= \sqrt{\frac{\sum_{i=1}^{n}(l_i-\bar{l})^2}{n-1}} \end{aligned} \tag{2}$$

In Fig. 8 and Fig. 9 the accuracy and precision are shown with respect to the distance and angle to the Hololens. A cubic interpolation is made between the measuring points. It can be seen that the best results are achieved between 40° - 60° and 0.7 mm and 0.9 mm. Here, the accuracy is 1 mm - 2 mm and the precision is 3 mm - 5 mm. If the angle decreases and the distance increases, it is not possible to detect the markers (white area). In addition, when angles become large, the accuracy and precision decreases strongly to max. values of 9 mm resp. 45 mm.

The determined accuracy is considerably worse than the pre-

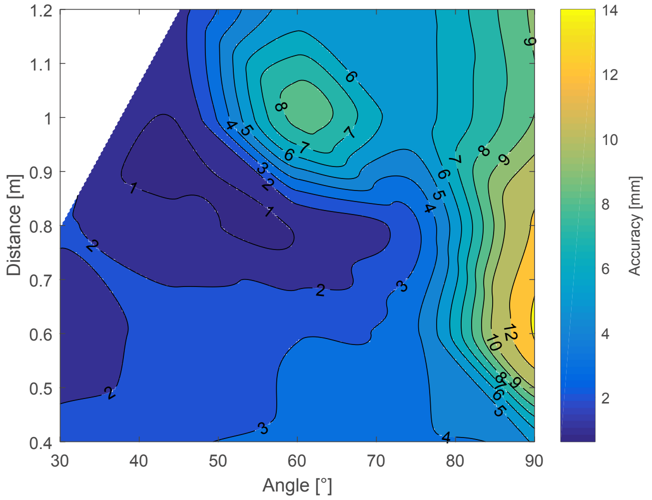

Fig. 8. Accuracy depending on distance and angle

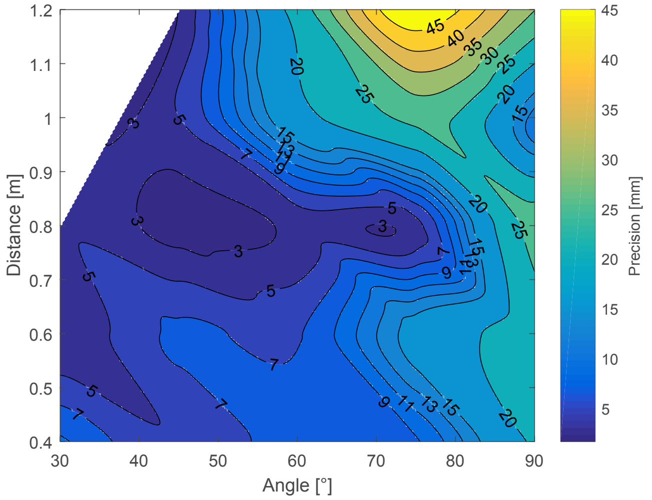

Fig. 9. Precision depending on distance and angle

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

cision stated by the robotic manufacturers (here: repeatabilty of Kuka KR 6: P_s=+/- 0.03 mm [24]). This accuracy was sufficient for this use case. Several tests have shown that the component could be gripped and set down correctly. This is due to the fact that the skirting boards center themselves on the mounting aid when they are placed on the mounting aid and thus bring themselves into the correct position.

6. Conclusion and Outlook

This paper is concerned with the development of a new Hololens application, so that a various components can be virtually assembled. By linking virtual and real components, an industrial robot can perform pick-and-place tasks based on start and end coordinates. This application has been implemented and successfully tested for two components as a sample application, but can be extended by using additional markers.

In order to determine the accuracy and precision of the marker recognition process, extensive measurements were carried out in this work. In the recommended scanning area an accuracy of 1 mm - 2 mm and a precision of 3 mm - 5 mm is reachable. During the testing of the system the components were gripped in almost all cases. However, this is a high degree of inaccuracy for robot applications.

In the following works, the accuracy of another marker tracking software, for example the *AR-Toolkit*, could be tested. On the other hand, the marker tracking could be replaced with an object recognition. This would allow an even more flexible and simple system structure. However, one marker for referencing the coordinate systems of the Hololens and the robot would still be necessary. Therefore it has to be evaluated if a higher degree of accuracy can be achieved by object recognition and the use of only one marker.

In order to further expand the robot's programming options, possibilities for describing a flow logic and path planning should be implemented. With the current application only a predefined pick-and-place sequence with a free adjustable offset is possible. A more flexible programming of the path can be realized by tracking path points from the virtual movement of the component. Based on this data, a robot path could be interpolated. In addition, assembly paradigms could be developed for various components from which gripping, approaching and placing modalities could be generated automatically. During this, the application should be extended to include the possibility of mounting components in all spatial directions. To do this, the coordinate transformation have to be supplemented by a transformation of the rotation coordinates.

The goal should be a dynamic generation of the robot program by the Hololens, which is sent to the robot. This would require the implementation of a different type of connection between the robot and the Hololens, since the current RSI only allows data exchange.

References

[1] Skovgaard, A.R.. Facilitating Programming of Vision-Equipped Robots through Robotic Skills and Projection Mapping. 2016. doi:`10.5278/vbn.phd.engsci.00062`.

[2] Pan, Z., Polden, J., Larkin, N., Van Duin, S., Norrish, J.. Recent progress on programming methods for industrial robots. Robotics and Computer-Integrated Manufacturing 2012;28(2):87–94. doi:`10.1016/j.rcim.2011.08.004`.

[3] Biggs, G., Macdonald, B.. A Survey of Robot Programming Systems. Proceedings of the Australasian conference on robotics and automation 2003;:1–3doi:`10.1109/ROBOT.2001.932554`.

[4] ABB Robotics. The future of robotics and automation depends on humans and robots working together. 2016.

[5] Univeral robots, collaborativ robots benefits. 2017. URL: `https://www.universal-robots.com/products/ur-robot-benefits/`; (accessed 5 Nov. 2017).

[6] Billard, A., Calinon, S., Dillmann, R., Schaal, S.. Robot Programming by Demonstration. Springer Handbook of Robotics 2008;:1371–1394,doi:`10.1007/978-3-540-30301-5\_60`.

[7] Landa-Hurtado, L.R., Mamani-Macaya, F.A., Fuentes-Maya, M., Mendoza-Garcia, R.F.. Kinect-based trajectory teaching for industrial robots. In: Pan-American Congress of Applied Mechanics (PACAM). 2014,.

[8] Lambrecht, J., Krüger, J.. Spatial programming for industrial robots: Efficient, effective and user-optimised through natural communication and augmented reality. In: WGP Congress 2014; vol. 1018 of *Advanced Materials Research*. Trans Tech Publications; 2014, p. 39–46. doi:`10.4028/www.scientific.net/AMR.1018.39`.

[9] Reinhart, G., Munzert, U., Vogl, W.. A programming system for robot-based remote-laser-welding with conventional optics. CIRP Annals-Manufacturing Technology 2008;57(1):37–40.

[10] Azuma, R.. A survey of augmented reality. Presence: Teleoperators and Virtual Environments 1997;6(4):355–385. doi:`10.1.1.30.4999`.

[11] Fraunhofer-Anwendungszentrum Industrial Automation, . smartfactory-designspace. 2017. URL: `http://www.smartfactory-owl.de/designspace/designspace.html`; (accessed 5 Nov. 2017).

[12] Elia, V., Gnoni, M.G., Lanzilotto, A.. Evaluating the application of augmented reality devices in manufacturing from a process point of view: An AHP based model. Expert Systems with Applications 2016;63:187–197. doi:`10.1016/j.eswa.2016.07.006`.

[13] Sacco, A.. Google glass takes flight at boeing. 2016. URL: `http://www.cio.com/article/3095132/wearable-technology/google-glass-takes-flight-at-boeing.html`; (accessed 26 Oct. 2017).

[14] Funk, M.. Augmented Reality at the Workplace : A context-aware assistive system using in-situ projection 2016;.

[15] Sand, O., Sebastian, B., Paelke, V., Carsten, R.. Virtual, Augmented and Mixed Reality. 2016. doi:`10.1007/978-3-319-39907-2`.

[16] Makris, S., Karagiannis, P., Koukas, S., Matthaiakis, A.S.. Augmented reality system for operator support in human–robot collaborative assembly. CIRP Annals - Manufacturing Technology 2016;65(1):61–64. doi:`10.1016/j.cirp.2016.04.038`.

[17] Danielsson, O., Syberfeldt, A., Brewster, R., Wang, L.. Assessing Instructions in Augmented Reality for Human-robot Collaborative Assembly by Using Demonstrators. Procedia CIRP 2017;63:89–94. doi:`10.1016/j.procir.2017.02.038`.

[18] Fang, H.C., Ong, S.K., Nee, A.Y.C.. Interactive robot trajectory planning and simulation using augmented reality. Robotics and Computer-Integrated Manufacturing 2012;28(2):227–237. doi:`10.1016/j.rcim.2011.09.003`.

[19] Chong, J.W.S., Ong, S.K., Nee, A.Y.C., Youcef-Youmi, K.. Robot programming using augmented reality: An interactive method for planning collision-free paths. Robotics and Computer-Integrated Manufacturing 2009;25(3):689–701. doi:`10.1016/j.rcim.2008.05.002`.

[20] Pai, Y.S., Yap, H.J., Singh, R.. Augmented reality – based programming , planning and simulation of a robotic work cell 2015;229(6):1029–1045. doi:`10.1177/0954405414534642`.

[21] Brosvision, . Augmented reality marker generator. 2017. URL: `http://www.brosvision.com/ar-marker-generator/`; (accessed 10 Nov. 2017).

[22] ISO 9283: Manipulating industrial robots - Performance criteria and related test methods. 1998.

[23] Weichert, F., Bachmann, D., Rudak, B., Fisseler, D.. Analysis of the accuracy and robustness of the Leap Motion Controller. Sensors (Switzerland) 2013;13(5):6380–6393. doi:`10.3390/s130506380`.

[24] KUKA Roboter GmbH. Spez KR AGILUS sixx. 2013.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.