Robotics and Computer Integrated Manufacturing 68 (2021) 102057

Contents lists available at ScienceDirect

Robotics and Computer Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Full length Article

Augmented reality integration into MES for connected workers

Andreea Blaga, Cristian Militaru, Ady-Daniel Mezei, Levente Tamas[1],*

Technical University of Cluj-Napoca, Robotics and Nonlinear Control Group, Dorobantilor 71, Cluj-Napoca, 400609, Romania

A R T I C L E I N F O

Keywords:
Augmented reality
Manufacturing execution system
Robotics

A B S T R A C T

In this paper an overview of recent augmented reality (AR) solutions for manufacturing execution systems (MES) is presented. The first part of the paper describes the challenges of integrating AR into MES, while the second part focuses on custom AR solutions. The last part of the paper highlights the advantages of the proposed approaches, as well as real life experimental results. Experiments are described in detail while the code for these applications is made public on author's website.

1. Introduction

The main challenges of the current manufacturing era are related to the high industrial performance requirements with relatively fast production life cycles and severe environmental constraints. The source of these problems relates to the loose vertical integration of digital trends within a manufacturing system with the natural demand for flexibility. One of the key requirements of the manufacturing execution systems relates to the data visualisation, for which the use of augmented/virtual reality devices for industrial workers [1] could be an effective solution.

The augmented reality term [2] refers to the addition of virtual objects to real world scenes in order to extend the capabilities of scene visualisation. To achieve this, one of the main challenges of an AR system is to align (spatially register) the objects of the real scene with the virtual ones. Besides the entertainment market, AR in tourism, real estate, marketing and remote maintenance is also playing an important role [3]. Even though in the last 50 years the AR technologies emerged continuously, their integration into the industrial applications is still limited. This integration accelerated in the Industry 4.0 context [4,5], mainly due to the remote assistance and maintenance applications[6]. Another boost for the integration was due to the use of digital twins for visualisation in industrial applications. Thus, the AR technologies have a great potential for manufacturing execution systems (MES).

Several benefits of the AR systems have been already proven recently to be of major impact in MES, including safety, precision, and learning curve of the operators. From the safety side, according to the International Labor Organization, every minute a work accident is happening. Thus, every step in the direction of reducing this trend is of major impact on the human workforce. As AR has a positive impact on safety in MES, this can reduce the rate of accidents [7].

In terms of Key Performance Indicators (KPIs), the increased precision and reduced assembly time with AR for safety critical aerospace industry was already proved in the 90's [8]. Thus, AR applications are favourable for optimising the production indicators on a large scale MES [9].

From the operators' training point of view, a recent study [10] highlights the fact that, with the introduction of the AR systems, the learning curve got enhanced for the remote operators, and first-time fix rates increased by 90%. For the assembly or remote assisted training applications, the impact of introducing AR solutions into MES has another major benefit: suggestive 3D visualisation is appropriate for the human perception. An example of such a visualisation is visible in Fig. 1, containing a partially assembled light bulb in the workspace of a collaborative robot, augmented with the final view of the same object (marked with red).

The integration of the AR techniques in MES is expected to have an overall positive impact. In this paper, the authors highlight some recent AR trends within MES, as well as describing use cases with different AR devices/technologies focusing on the benefits of the integration of these tools. These were successfully integrated into customer specific applications. Section 2 focuses on the main literature related to the AR integration into MES, while Section 3 presents typical applications in the field of remote assistance, remote navigation, robot state monitoring and collaborative assembly. These applications got better by integrating AR technologies: shorter reaction time on specific events from the connected workers side; more intuitive information representation for different processes; reduced collaborative assembly time with a cobot and a safer interaction with an AGV in a production system.

* Corresponding author.
E-mail address: Levente.Tamas@aut.utcluj.ro (L. Tamas).
[1] All authors have contributed equally

https://doi.org/10.1016/j.rcim.2020.102057
Received 20 December 2018; Received in revised form 31 August 2020; Accepted 1 September 2020
Available online 11 September 2020

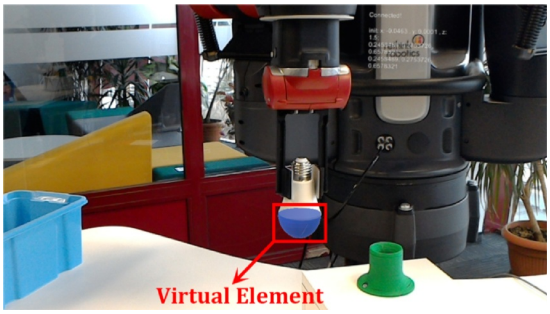


Fig. 1. Preview of final product with augmented visualisation [11].

2. Related work

Traditional manufacturing focuses on ensuring product quality, but nowadays clients take quality for granted also in case of emerging tech products. In order to stand out, the business process must adapt and provide exceptional service, by the means of well-defined mechanisms. The main challenges of modern manufacturing are: individualisation (product customisation by client requirements and needs), adaptation to market fluctuations, and the need of a strong networking with suppliers and vendors [12]. By introducing digital MES flexibility, transparency, responsiveness and cost efficiency was achieved. This is a compact integration of classical techniques: production data acquisition, staff work time logging, quality assurance and finite scheduling. In this way, MES allows fast response to events (e.g. production downtime) and changing requirements [13].

Augmented Reality (AR) has a long history in the researcher community, focusing on visualisation technologies. This type of visualisation ensures the mixing of real and synthetic information in the same view.

Usually, AR systems are realised using a head-mounted display (HMD), which can have different instances such as glasses or helmets. Besides the technical realisation, AR systems can be characterised by: real time user interaction, combining real and virtual and aligning the elements properly in a scene. With these characteristics, simple devices such as smart phones or specific projectors can be considered as AR devices as well [14]. The most important aspect of the AR devices is related to solving real life problems, such as remote maintenance or operator training for manufacturing applications in the context of Industry 4.0 paradigm.

2.1. Current AR trends in MES

MES is a clear need for product quality assurance, along with overall productivity. In the Industry 4.0 era commonly used tools such as data management, simulation and visualisation boost productivity [3,15].

In recent years this notion is thought of being a part of innovative technologies, while talking about diminishing cost and time for product development. The main concerns are customisation requirements, quality, and market reaction time [13,16]. Smart manufacturing has its focus on a product's life cycle and its associated data, while aspiring to develop flexible manufacturing processes able to react to swift changes. Information throughout the process is available at any time for the entire network. Associated fundamental technologies involve human-machine interaction and robotics as well [17–19].

The idea of using augmented reality (AR) in industry is not new: already in the90′s the aerospace assembly industry used it in order to increase precision in production [8]. Recently, AR within the industrial context is discussed in the review paper [20]. A good starting point for the taxonomy analysis of the AR systems is available in [21], while details regarding the specific manufacturing domain is presented in [22] and [3]. The latter presents some AR platforms for MES such as glasses, head mounted devices or tablets, discussing the tracking approaches for these devices in part.

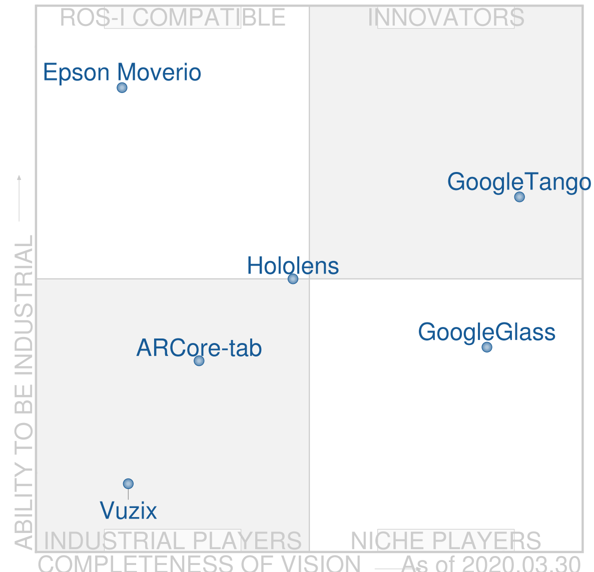


Fig. 2. ROS based AR device magic quadrant.

In the classification suggested by the authors of [3], field and maintenance applications are distinguished. From the field applications view, the most relevant ones are from the aviation industry, remote applications and worker safety training. The maintenance related applications are focusing on repair, diagnosis and remote operator training domains. Furthermore, AR solutions are also suited for the safety critical industrial applications such as nuclear plants, power plants and aeronautical industry [7].

An important constraint for AR devices in the industrial context is the ability to integrate with open source, e.g ROS industrial based packages. In our investigations we focused on the devices which have some already existing open source packages, or we could implement ourselves within a short period such an interface. We also took into account the innovative solutions implemented into the device, as well as the maturity of the available documentation. According to our classification criteria, the magic quadrant for the list of devices is as in Fig. 2.

2.2. AR integration into MES

In numerous industrial processes, the human worker is an elementary part of the assembly scenario. Human intuition is essential in some operations. However, the unpredictability of the human operator makes challenging coworking with robots [23,24]. The collaboration between human and robots needs to be taken into consideration for having promising benefits. Combining a robot's robustness, strength and precision with human perception, insight and flexibility brings inestimable features for scaled-down production in particular, as shown in Fig. 1.

Safety is a main concern in human-robot collaboration. Powerful manufacturing robots are constrained to work behind fences in order to protect the operators [25]. Lightweight robots are considered safer, but less efficient than industrial ones. SafetyEye [25] is a system that tracks operators and stops the robot if the danger zone is violated. Also, limiting force and speed is recommended during close human-robot collaboration phases.

By using AR technology with human-robot collaboration for smart manufacturing is a beneficial scenario which combines key traits of the used technologies. By using virtual elements to enhance perception blended with robotic characteristics (sturdiness and accuracy), the industrial processes become explicit environments that efficiently manage products' life cycles, but also cut down cost and encouraging teamwork

[26].

In smart manufacturing, AR is one of the cutting-edge technologies that may be used to reduce the execution time of tasks, thus production and labour cost is significantly reduced. However, various vital concerns must be taken into account when talking about these operations that may instead decrease productivity. Human error is the main issue, since difficult tasks are subject to misguided directions to complete. Incomplete training also plays an important part in this [27]. Having used AR, any erroneous activities are significantly rectified due to the visualisation in production of the distinctive stages. In addition, a noteworthy benefit is the possibility of simulating the manufacturing processes when taking into consideration the real equipment cost. There is no damage to the components since all actions are done in virtual settings.

The development of typical manufacturing scenarios using AR technology brings enhancement of human perception by working with a set of elements for common industrial tasks, exposed through an interface. Having such an environment, distinguishing data needs to be provided for a specific scenario at convenience time and place so that the wanted output is obtained.

The specified characteristics are to be incorporated into applications specialised in production, by outlining and validating the order of operations to execute, considering a customised model. By using AR, assembly data and instructions are graphically designed and displayed when needed. At the same time, its corresponding sequence is superimposed in the virtual setting on the authentic product [28]. Continuous calibration must be implemented so that the real and virtual parts remain spatially aligned in an unitary format.

In order to detect relevant items in real-time, an image-based solution can be used. Visual markers act as an anchor between real and virtual elements as they are placed in the 3D space and are camera identifiable. Predefined patterns (such as QR tags) are used for recognition, from which the camera position and orientation (referred as external calibration parameters [29]) can be extracted later on. Using the camera parameter estimate from this step, the re-projection of the augmented content over the real image is performed, yielding to measurable overlap difference, often denoted as delta error [30]. Therefore, camera parameters are a decisive factor for accuracy of see-through devices [11].

3. AR integration into MES with a cobot

The main objectives of this paper is demonstrating possible scenarios for human-robot collaboration enhancements when talking about assembly manufacturing execution systems (MES). AR in assembly tasks is a research area of growing interest. As [31] shows, AR is suitable for multiple stages of the assembly process. AR has been successfully integrated in assembly guidance, assembly training and assembly planning & design. This article focuses on assembly guidance, i.e. displaying relevant assembly information to the operator. AR is integrated with the robot by developing a solution that is capable of communicating with the robotic workspace, as well as sending and receiving information about the position of the worker and robotic key points.

When talking about assembly, a physical item can have a digital twin, which is a CAD model that accurately simulates the real characteristics [32]. This 3D representation is made of several components with a specific hierarchy, so that a beneficial assembly sequence can be determined.

Multiple issues may arise while developing AR applications applied to industrial scenarios, especially concerning objects without distinguishable visual patterns or with shiny faces. These kind of features may lead to imprecise 3D pose estimates [30]. In addition, an environment with certain lighting may greatly limit accuracy. A solution is working with leading equipment, like the Microsoft HoloLens, as these devices can better control sudden ambient light variations [33].

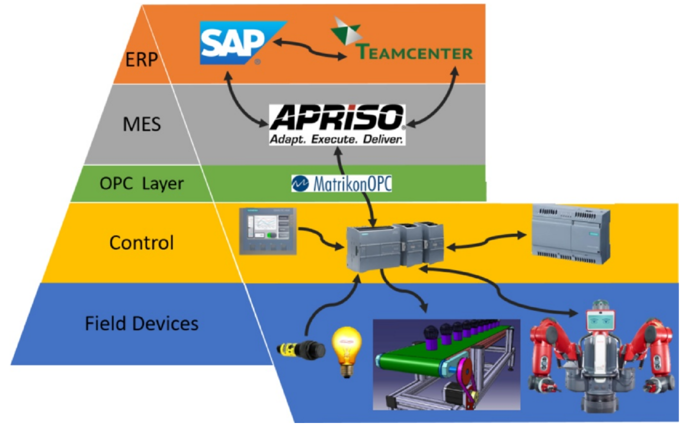


Fig. 3. MES with AR System architecture overview.

For larger scale application the use of Google Tango is more appropriate, on close object inspection ARCore proved to be more efficient [11], while Vuforia library proved to be an easily extensible approach. Although the Google Tango project officially is not supported anymore, the mature documentation and robust tracking features of this approach made it an appropriate choice for AR applications.

A conceptual overview of the AR integration into an existing MES is presented in Fig. 3. In this architecture on the shop floor the following control and monitor equipments are involved:

- Siemens S7-1200 PLC
- Siemens KTP400 Touch Panel HMI
- Siemens IoT2040 intelligent industrial gateway
- Built-in Baxter controller
- Conveyor belt
- Proximity sensor
- AGV from Turtlebot company

At the software side the following elements were integrated [13]:

- Solidworks to design the CAD model for light bulb holders and tester box
- Siemens TIA Portal V13 for PLC and HMI programming
- Node-RED for Siemens IoT2040 and IBM Bluemix
- Robot Operation System (ROS) for programming the Baxter cobot and the AGV
- DELMIA Apriso MES solution
- SAP ERP solution
- Teamcenter PLM solution
- IBM Bluemix as cloud service provider

This setup is proposed for a client use case highlighting the benefits of open source solutions, including emerging ones such as AR technologies in the field of MES. Further details regarding the plant setup can be found in [13].

3.1. MES application overview

Our aim was improving day-to-day manufacturing operations by developing human-robot collaboration systems with augmented reality. One of the main goals is the 3D vision enhancement of industrial operators with the help of head-mounted display (HMD) equipment, which eases instantaneous perception of the scenario.

Having used augmented reality, virtual elements for the assembly sequence of the final product can be visualised before beginning the actual production, as it can be seen in Fig. 1. On the one hand, for the design, the assembly model of the product can be decomposed into a characteristic hierarchy with definite assembly sequences. On the other hand, for the production, the industrial worker can obtain relevant visual information with the use of these sequences with the possibility of adaptation to the current stage of the assembly. The overall development of the product can considerably prosper using this system, resulting to a significant decrease in lead time (initiation - execution latency), cost reduction and quality improvement [34].

When talking about visualising a 3D assembly scenario, the first phase is the real-time synchronisation of real and augmented elements in the shortest time possible. We need to take into account the constant need for space calibration so that the real and virtual scene are kept aligned in space for the final stage of one, stable system [35]. For this reason, one of the methods that can be used to identify essential elements and their position is the marker method. With this, HMD calibration can be done as well. Furthermore, the human-robot collaboration setting needs to be tuned by calculating the structural connection between the robotic space and device to be operated on. Thus, accurate links between the position and orientation of involved equipment in the scenario are obtained.

In order to develop an AR application that collaborates with a robot, an important step is taking into consideration the different coordinate systems for both technologies and their point of origin in the real world. In these circumstances, the robot may get imprecise data regarding the operator's place. In turn, the operator can receive misleading information related to the robotic workspace. To straighten this out, the correct spatial transformations needs to be computed between the position of virtual items and the workspace of the robot. It is worth taking into account human-robot collaboration safety matters, where attention and alertness of humans is unsatisfactory.

Implementing AR applications require dedicated software and hardware. Specialised equipment like HMDs and proper trackers must be attained [22]. When considering hardware, an appropriate wearable device is required so that is favourable for the entire system. Characteristics such as size, weight and portability are considered when talking about manufacturing tasks, which needs to assist the industrial worker all day. In order to avoid severely reducing data that can be visualised at once, a convenient field of view should be considered [36].

3.2. Hololens integration into MES

For our application involving the Hololens AR HMD, the focus was on the integration into a custom collaborative robot application built using Robot Operating System (ROS). The current pose of the HMD and the pose of the objects in the workspace of the robot are identified using visual markers. This pose data is transformed in ROS messages and then used as frames for the transform (tf) tree. Within ROS, with a predefined inverse kinematic model, we can compute the position of the end-effector used for the assembly scenario. This information is needed in order to accurately augment the next element to be assembled for the final product, as it may be seen in Fig. 1.

First of all, the camera of the HoloLens needs to be calibrated so that the finest tracking accuracy is obtained, specifically when glancing straight into a flat marker. For this type of device, lens distortion in the video image that is displayed may be removed as well. In order to accomplish this, the tool ARToolKit [37] was combined with camera calibration from OpenCV [38]. Parameters of the camera may be found using explicit calibration algorithms [11].

With ARToolKit and a see-through device, in this case HoloLens, virtual items can be superimposed in the real setting. Computing positions of virtual elements in the 3D space is done by identifying and tracking square markers in the real space. Camera view video is captured by the camera and sent to the software which searches for fiducial markers in each frame of the video. If any with the pattern is identified, the software then calculates the square's position, as well as the orientation of the pattern, considering the camera. The described algorithm may be visualized in Fig. 4. After this data is known, a virtual model with external camera calibration is drawn, as it can be seen in Fig. 5.

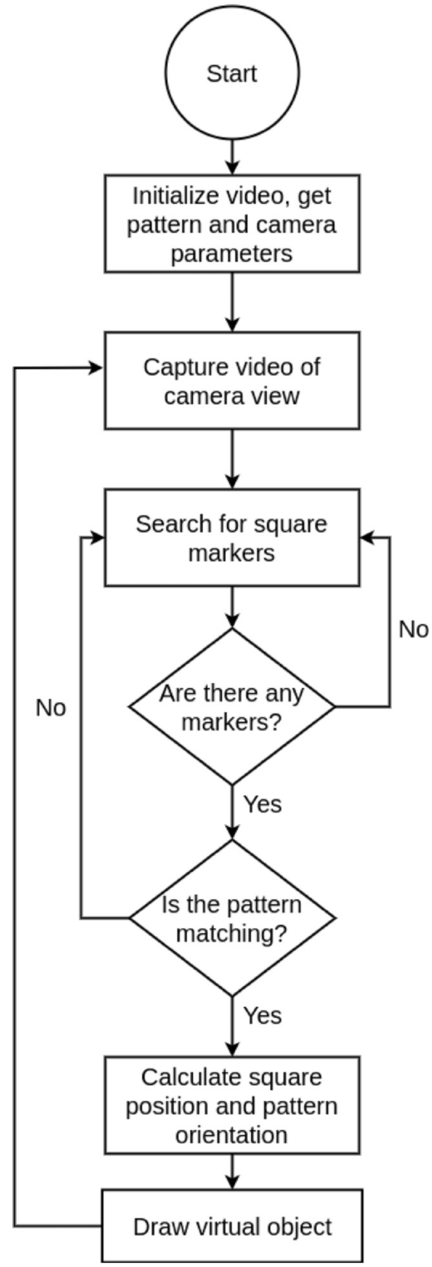


Fig. 4. HoloLens calibration algorithm.

For the ARToolKit with HoloLens integration, a Universal Windows Platform (UWP) wrapper was used. It also contained the marker recognition and tracking algorithms. Knowing the camera parameters and the pattern of the marker, the accuracy of the parameters can be verified. A custom virtual item can be visualised. The marker recognition and tracking procedures found in the kit may determine the position of the object.

The HoloLens application is communicating with the robotic workspace through a common protocol with the Rosbridge package. This yields to a UWP–ROS interface, which can be relevant in particular for non-ROS applications, as one can visualise in Fig. 6. In order to send data from HoloLens to ROS, custom C# classes were created so that they can simulate the message type that needs to be received by the ROS service. For this, Hololens' current position and orientation is sent, as well as the transform of the initial marker for the pattern that is tracked. To receive this, the service callback gets the data and, with the help of a ROS publisher, it publishes the information on the /tf topic.

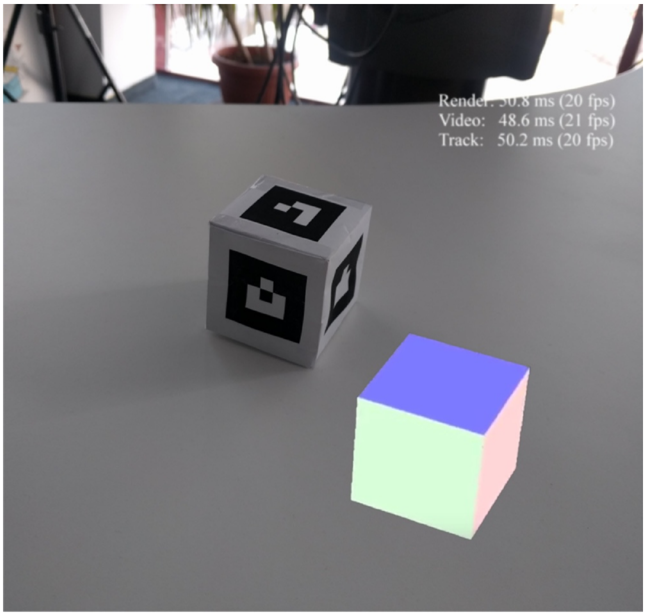


Fig. 5. Visualisation of the real object and the AR marker during calibration [11].

The transform between the robot's right end-effector and the HoloLens' initial position is computed, resulting in the element of the robot transform with respect to the device's coordinate system. This transform is the response for the HoloLens application. The transforms can be seen in Fig. 7.

3.3. Hololens and cobot external calibration

HoloLens development system using Unity3D has a left-handed coordinate system. ROS works with a right-handed coordinate system. In this situation, the conversions between left-handed to right-handed systems has to be computed.

In order to finalise the conversions, further calculation needs to be done so that the HoloLens' system is represented in the robotic coordinate system. This can be written as:

$$T_H^G = T_H^X \cdot T_X^G$$

To convert from ROS coordinate frame convention to the one used by HoloLens, the inverse computations need to be addressed. This is defined by transform multiplication:

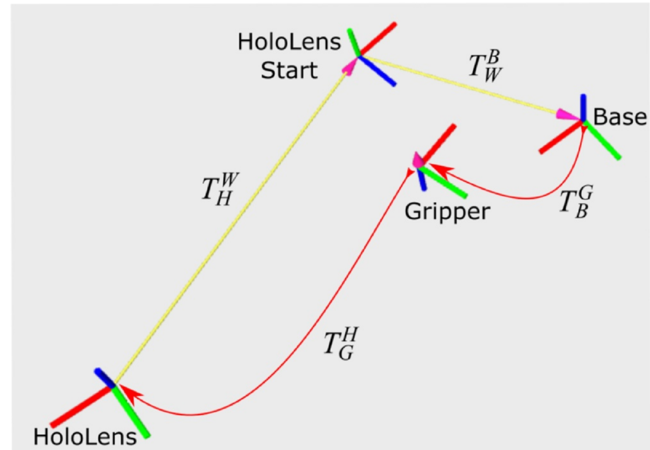


Fig. 7. Frames used in the application [11].

$$T_G^H = T_G^Z \cdot T_Z^H$$

The summary of the symbols used during the calibration can be found in Table 1.

The extra step needs to be done here as well, where the right-handed HoloLens is computed to the left-handed classic one, the same inversion type as mentioned above.

The external calibration between the HoloLens HMD and the dual-arm robot is a hand-eye calibration type procedure [39]. This is done by estimating the spatial relation between the two coordinate frames using artificial visual markers. This was achieved by attaching the camera to the robot arm, and recording several arm-camera data pairs and from this estimating the cross calibration between them [40]. For this to be feasible, the HoloLens' position considering its start position is already known (device having motion tracking), the start position does not change in relation to the base of the robot. The end-effector pose is also prior computed, with respect to the base. In order to find the unknowns, multiple samples are used. These are gathered by having the end-effector move in a variety of positions and angles, managing to cover a large range.

Having a multitude of samples that were measured, external camera calibration [41] can be greatly improved, in such a way that the virtual element may be precisely spatially aligned with the real items.

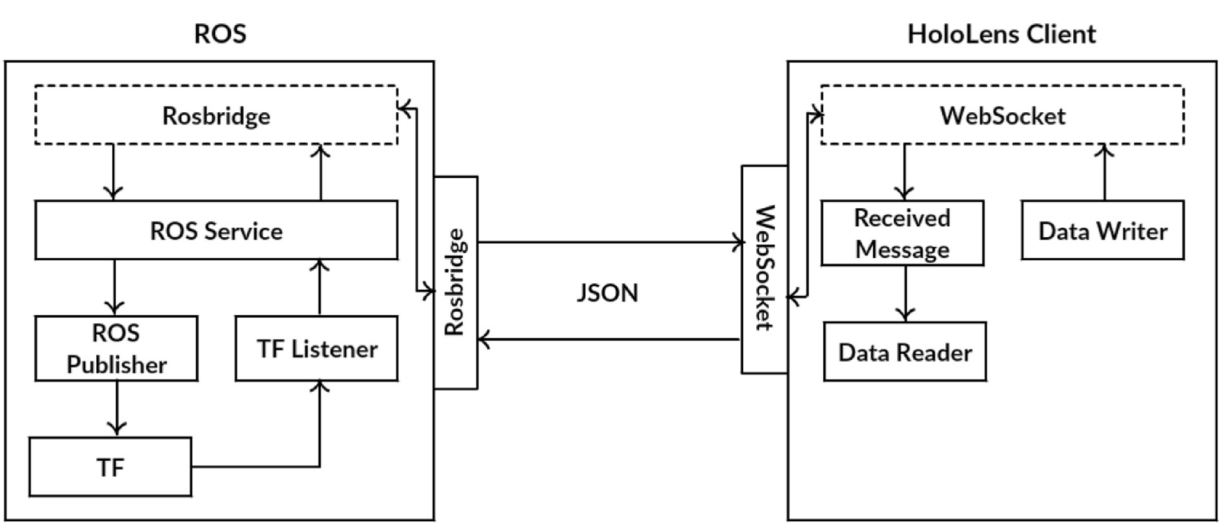


Fig. 6. Rosbridge communication between cobot and AR device [11].

Table 1
Transformation symbols meanings.

Symbol	Meaning
T_H^G	Transformation of HoloLens coordinate system in robotic coordinate system
T_H^X	Rotation of HoloLens' coordinate system on X with − 90°
T_X^G	Rotation of HoloLens' coordinate system on Z with 90°
T_G^H	Transformation of robotic coordinate system in HoloLens' coordinate system
T_G^Z	Rotation of robotic coordinate system on Z with − 90°
T_Z^H	Rotation of robotic coordinate system on X with 90°

3.4. Google Tango and AGV external calibration

For the path visualisation application (Section 4.1.1), a different calibration algorithm was used, as this is particular to each robot used in the experiments.

In case of the Google Tango AR device, the self-localisation was not performed accurately enough, so recalibration was needed before each experiment. In order to speed up the external calibration procedure, a straightforward method was chosen, described next.

Both devices (AR device and mobile robot) run motion tracking algorithms, each one having its own coordinate system. To create a cross link between the two coordinate systems, the AR device is placed on top of the robot, in a predefined position. Then, the computation of a rigid homogeneous calibration transform is done, as follows:

$$T_{ARW}^{RW} = T_{ARW}^{AR} \cdot T_{AR}^{R} \cdot T_{R}^{RW}$$

where *RW* stands for Robot World coordinate frame, *R* is the Robot, *AR* is the AR Device and *ARW* is the AR World. T_{ARW}^{AR} is the pose of AR device, obtained from motion tracking. T_{AR}^{R} is constant, measured manually. T_{R}^{RW} is inverse of robot pose, from motion tracking.

Using this method, a transform from *Robot World* to *AR World* is obtained, with absolute precision up to 10cm in the robot coordinate frame system, the radius of the robot being a half meter.

4. AR based visualisation use cases

In this part we summarise our use cases, divided into two different categories: the first focusing on the robot state visualisation using AR devices, while the second ones are presenting collaborative assembly scenarios.

4.1. Robot state visualisation

4.1.1. Path visualisation application with Google Tango

The main goal is to improve the safety of workers and an automated guided vehicle (AGV). In this section, an application for visually inspecting robot's path is described. In this application, AGV's computed path is overlayed live with the images of the environment. The user checks if the drawn path intersects any obstacles and, in that case, stops the AGV, thus leading to a safer interaction with the autonomous agent. A demonstrative video can be seen on the authors' webpage.

In this implementation, existing technologies are reused. We use TurtleBot as AGV, which has a navigation stack available in ROS [42]. This includes motion tracking, planning and executing paths. Regarding the AR device, Google Tango [43] technology powered device is chosen, having motion tracking and augmented reality technologies out-of-the-box. The two devices communicate with wireless network using ROS framework, with additional Android support for the AR device. Although the Google Tango project officially is not supported anymore, the mature documentation and robust tracking features of this approach makes an appropriate choice for an AR application.

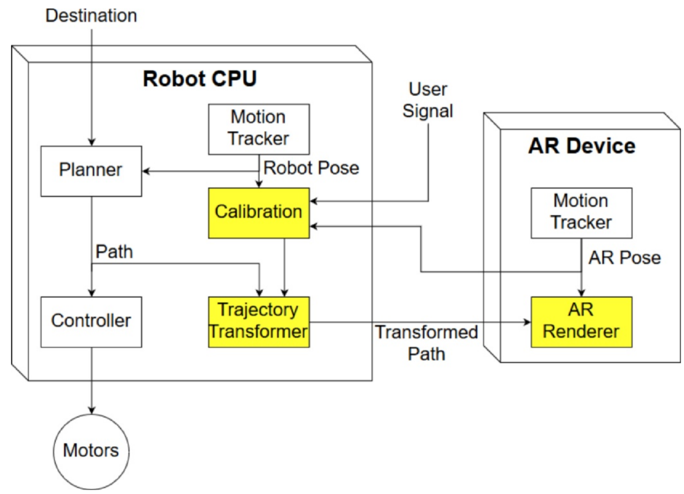


Fig. 8. Path visualisation application overview.

Fig. 8 shows the main components of the application, highlighting our contribution. That is an external calibration algorithm, described in Section 3.4, as well as a rendering procedure. Experiments were performed in order to determine the quality of AR visualisation.

In this case, the planner produces a sufficiently large number of waypoints, with small distance between them. Therefore, the AR device has a low frame rate. To overcome the issue, the Douglas-Peucker smoothing algorithm is employed [44], which reduces the number of points on a path degrading the original curve up to a preset threshold.

In the experiments, the distance between the AGV's real position and the one shown on AR device is manually measured. The initial error (measured right after calibration) is, on average, 0.1*m*. After the robot travels 10*m*, the accumulated mean total error in is less than 0.2*m* (the robot diameter). Hence the AGV can be used in a safe manner for path planning and visualisation applications using at least a 0.2*m* clearance path.

4.1.2. Robot state visualisation with HoloLens

In the Industry 4.0 context is important to know what the robot coworker is doing in real-time. By visualising certain variables and parameters from the coworker, a better understanding can be gained. Thus, to the human operator a set of valuable visualisation tools are offered, with the main goal of enhancing safety and to reduce downtime in the production.

The application is based on a Microsoft HoloLens device and exposes data from the ROS environment running on the robot. The communication between the robot and the HoloLens HMD is done by the Rosbridge package, using messages in JSON format (Fig. 10).

The application is composed of three panels: connectivity, advanced and extended options panel. At startup only the connectivity panel is shown, the assists with the connection procedures with the ROS environment running on the Baxter coworker. If the connection is successful the advanced panel will pop up. The advanced panel contains functionality for sending/receiving information about the robot, such as information about the running nodes and published topics. The global state of the robot can be inferred from the information contained in this panel. There are three indicators that will change their colour in red or green, depending on the sub-modules that reflect the robot's activity state such as this is visible in Fig. 9.

In case of malfunction the advanced options panel will allow the user to perform safety stop, restart actions, etc. More precisely, the human operator can check the running state of a certain node and can stop/relaunch it individually as well.

More details regarding this application can be seen in the video.[2]

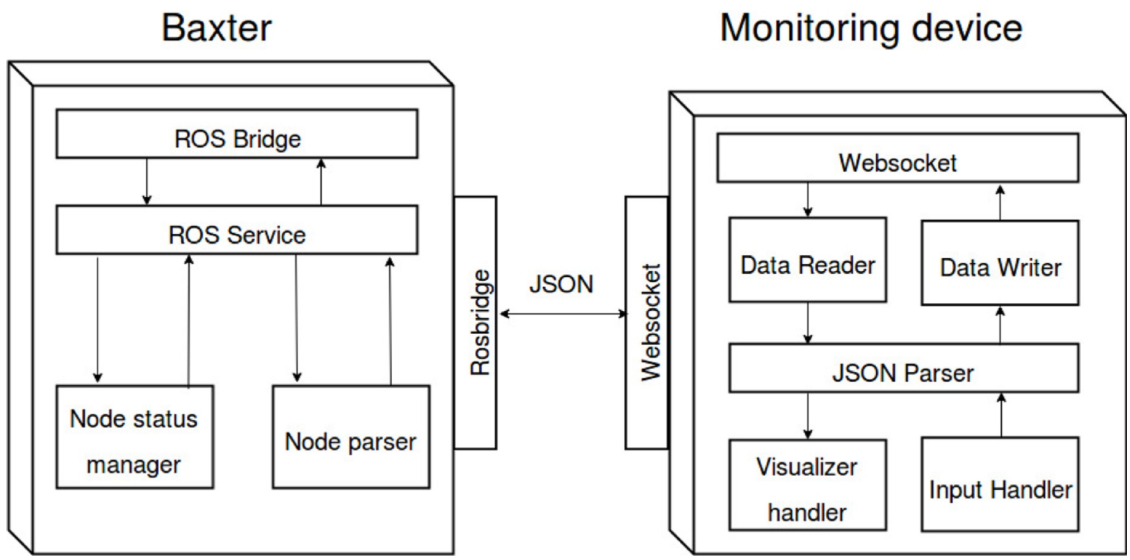


Fig. 10. The components of the monitoring tool.

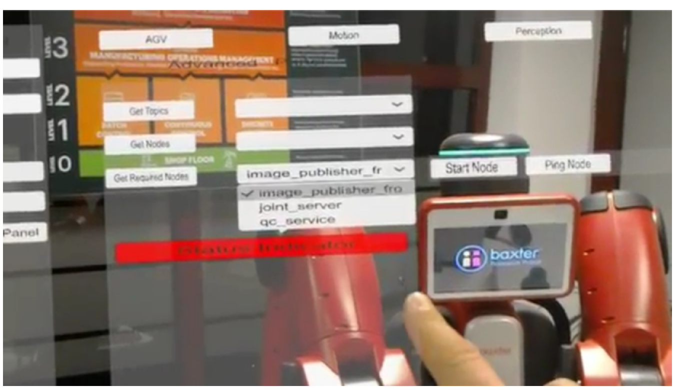


Fig. 9. Performing a status check using the monitoring tool.

For this application we relied on the default external calibration of the HoloLens device, which was sufficiently precise for visualisation purposes. The external calibration in this case is done using a static calibration method, i.e. with respect to a fixed point in the robot working space.

4.1.3. Robot state visualisation with Epson Moverio

An important aspect of interacting with robot teams is the state visualisation of the different devices. This is essential in the remote debugging or in the commissioning stage of the production line. In order to facilitate this problem, we propose an open source, ROS industrial based integration of AR devices using Android and Vuforia SDK. The developed package[3] contains a generic web interface, which allows to interconnect different types of robots and get online information about them. This information ranges from operational parameters to the online variables of the robot. This bridge allows communication with the Android enabled devices in an efficient way by using for instance JSON messages, as illustrated in Fig. 11. In order to identify the robot with the AR device, we used a pretrained database with the visual identifiers of each device. This is achievable with the user friendly SDK from Vuforia, which allows both online (through their web server query) and offline target recognition (using local custom database).

For the Epson Moverio BT-300 the overlap of the AR image with the real word depends very much on the initial internal calibration procedure, which can be performed using custom helper applications from Vuforia SDK.

In the real life experiment we focused on the visualisation of the required node list visualisation for different interacting robots (such as AGVs or robot arms). A demo video,[4] as well as the code, is available on the corresponding author's homepage.

4.2. Collaborative assembly use cases

4.2.1. Digital twin visualisation with HoloLens

The use of digital twin got widespread with the appearance of low cost 3D printing solutions. The real product that needs to be assembled is virtually cloned using CAD modeling. This procedure can precisely simulate the physical traits, the virtual model being considered a digital twin for the real one. This model was divided into several components so that a definite ranking may be determined for an assembly type application, as it can be seen in Fig. 12.

In order to execute the extrinsic calibration of the HoloLens camera, different algorithms were used so that calibration error is minimal. Using the camera-robot calibration package described in 3.4, when starting the application, the user had to change the pose of the HoloLens in the proximity of the robot. A marker is attached to right end-effector of the robot visible by the HoloLens for the cross-calibration. This method was determined to be inefficient due to being a burden on the user. Additionally, if marker detection fails in just one iteration, the end result would be poor. Instead, by attaching the device to the left end-effector,[5] the first task of the calibration is considerably improved, so that the calibration is better approachable, as seen in Fig. 13.

In the start phase of the final application, the robot's right end-effector holds the HoloLens, whilst visual marker is fixed on the left arm for the extrinsic calibration. Proper calibration gives precise synchronisation between the real and virtual elements, resulting in fitting spatial alignment. More than one iteration may be needed for this. After being finished, the industrial user may visualise the virtual components of the scene in their correct place, with respect to the coordinate system of the HoloLens. This is visible in the demo[6] video.

[2] https://youtu.be/3AR9hE6JKh0 .

[3] Demo available at https://ngi-systems.atlassian.net/wiki/spaces/ROS2AR/ .

[4] https://vimeo.com/351293877 .

[5] https://youtu.be/3AR9hE6JKh0 .

[6] https://www.youtube.com/watch?v=BjXEL-VZbnw .

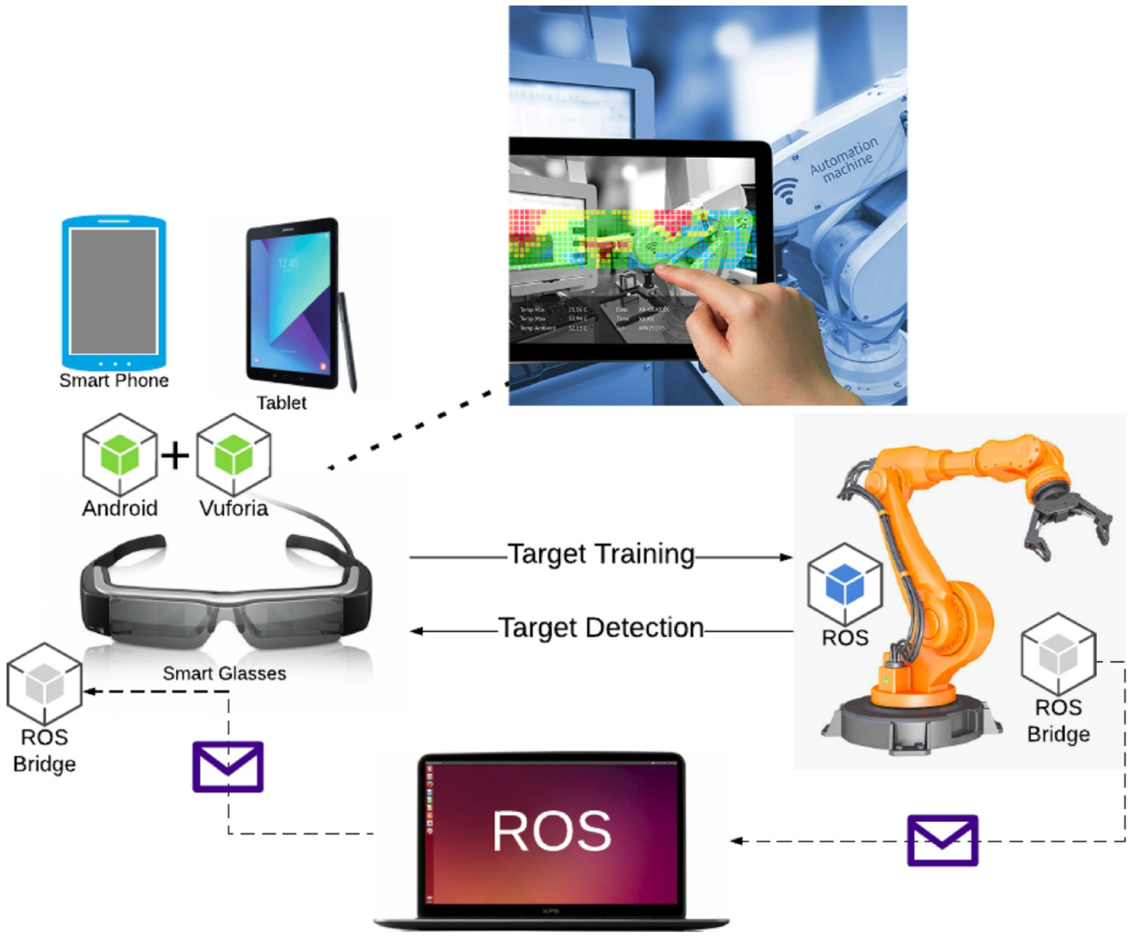


Fig. 11. The components of the Android robot state visualisation.

Fig. 12. CAD model with augmented part (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

We validated the relative pose estimate between HoloLens and the Baxter robot performed in the previous step. In order to be able to properly repeat the tests, they were done in three phases. In the first place, the right end-effector holds the device fixed, while the normal procedure of the calibration is executed. After, the HoloLens was fixed to the left end-effector, recording its position again. Next, the virtual component's transformation of the end-effector was compared with the real one, computing the degree in which the virtual element is overlapping its physical correspondence as seen in Fig. 14.

Fig. 13. Calibration procedure with Baxter and HoloLens.

After repeating the individual procedures several times, the data that was obtained was processed using two ways: the mean and the standard deviation. Firstly, each of the measured samples related to the device's position with respect to the end-effectors was used in order to compute their corresponding Eucledian distances. The resulting errors for each end-effector were used to determine the mean and standard deviation. As it may be visualised in Table 2, for the right end-effector, the measurements are considerably lower than the left one due to the calibration being done with the first. Next, the virtual element superposition on the real one was calculated taking into account the pixel number of the virtual which overlap the real component, converted into percentages. The final data may be seen in Table 4. For the HoloLens system, a $\approx$ 58.83% mean is determined, corresponding with the overlapping percentage, with $\approx$ 11.84% standard deviation. For the ARCore application, the determined mean is $\approx$ 78.32%, with $\approx$ 9.87%

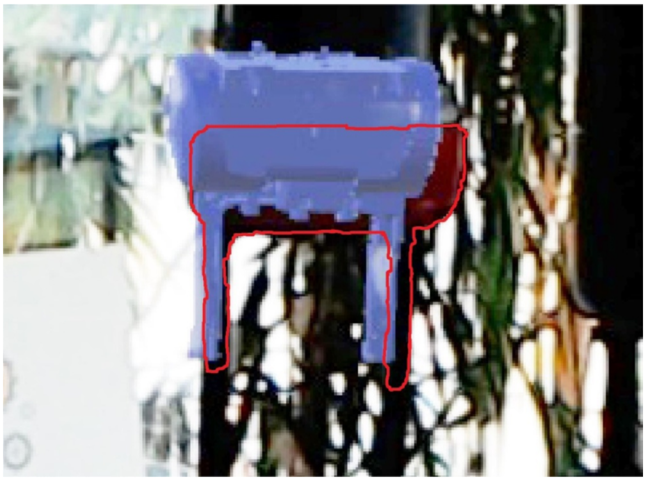


Fig. 14. Overlap between virtual object (blue) and real object (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
The output from the calibration process for the Baxter and HoloLens [11].

End-Effector	Error Mean (cm)	Error Standard Deviation (cm)
Right	0.47	0.19
Left	0.95	0.45

standard deviation.

Table 3 contains an overview of calibration methods used in this paper:

- In static calibration (Section 4.1.2), the AR device is always started from the same position. Therefore, the robot's position relative to AR device can be manually measured.
- The one-shot calibration (Section 3.4) provides more flexibility by allowing the user to start the AR device in any location, then place the device in a predefined position relative to the robot and calibrate the systems.
- The automated procedure (Section 3.1) increases the accuracy by taking multiple measurements. Optimisation algorithms [40] can compute the transform between the systems, having two tracked moving elements (AR device and robot end-effector), which are physically linked together.

After the calibration process is completed, the real assembly objects can be used simultaneously with the running application. At any particular moment, the industrial operator may visualise the matching element that needs to be assembled and receive data about assembly information, as well as its proper position and orientation. For this, the robot's right end-effector holds a part of the assembly, whilst the succeeding part may be displayed using HoloLens. By having the response data given by ROS, being the pose of the right end-effector with respect to the coordinate system of the device, the virtual object's pose is

Table 3
Calibration methods.

	Static	One Shot	Automated
Type	Manual	Manual	Manual + Automatic
Duration	Very fast (seconds)	Fast (seconds)	Slow (minutes)
Accuracy	Medium	Medium	High
No. of Measurements	0	1	More than 10
Flexibility	Low	Medium	High

Table 4
The augmented and physical object overlapping.

	Mean (px)	Standard Deviation (px)
Overlap - HoloLens [%] [11]	58.83	11.84
Overlap - ARCore [%]	78.32	9.87
Overlap - Moverio/Wikitude [%]	64.09	5.12

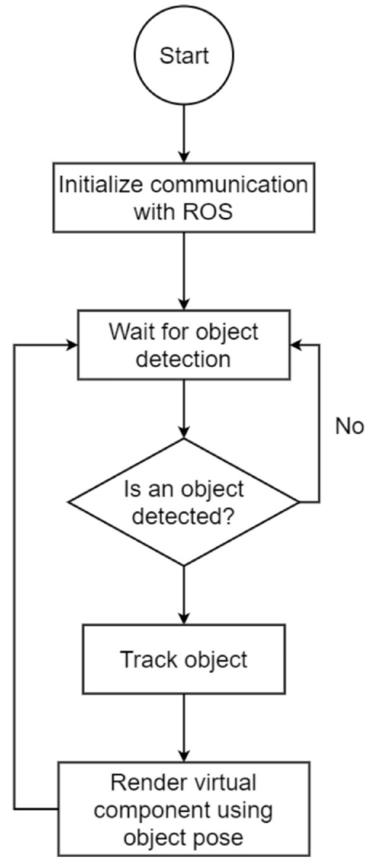


Fig. 15. ARCore application algorithm.

computed, taking into consideration the current pose of the physical element as well. The overall application algorithm may be seen in Fig. 15.

A modest alteration to the above mentioned scenario is that the robot may not hold any part of the assembly, the real objects being positioned only near the robotic workspace, so that the end-effectors of the robot can reach the product and still be able to aid the industrial operator in the assembly task. This application may be visualised in Fig. 17(b), showing a small scale assembly use case.

4.2.2. Small scale assembly with AR Android device

For the ARCore system, experiments were done as well, but in a fairly different fashion for some points. At the start of the application, after detecting and getting into the tracking regime, the virtual element is superimposed onto the real scenario. Thus, the degree of overlapping for the virtual unit over the real one can be computed, similar as above. For this particular scenario, an Android enabled AR-capable mobile device was used, having integrated the ARCore SDK in the application.

The communication among the Android and ROS ecosystem was done by using the ROSJava package [45], which allows a Java client to connect to a ROS node, facilitating the ARCore integration into ROS.

At the start of the application, the user is prompted to enter the URI of the ROS master. In order to exchange information between ARCore and ROS, a custom service client was implemented and constructed

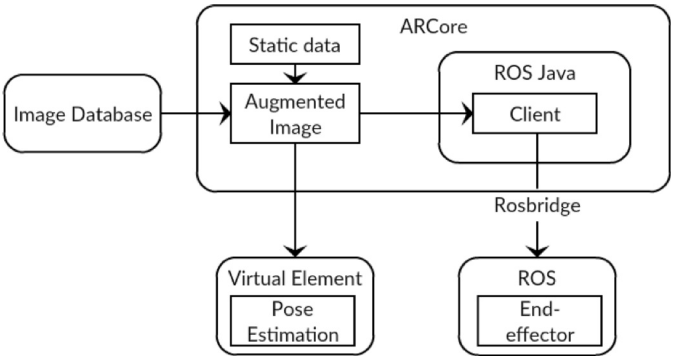
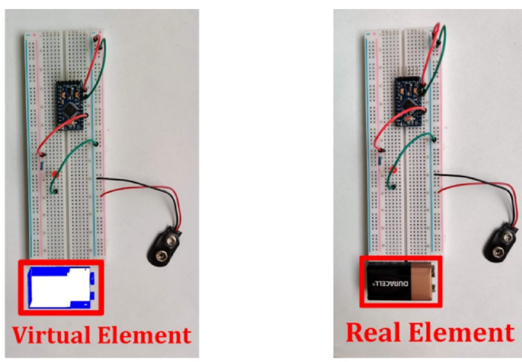


Fig. 16. ARCore application architecture.



(a) Task augmentation snapshot (b) Application outcome

Fig. 17. ARCore application.

such that its service type simulates properly ROS' resources. The pose of the camera device for every frame is the data sent to ROS, along with the currently identified and tracked elements from the scene.

The overall architecture may be visualised in Fig. 16.

For integrating ARCore in the mentioned scenario, an already implemented sample from the ARCore SDK was used as a base for the application. In the first phase, the image database was recreated using fitting images with different stages of assembly for the final product. In this case, an electronic circuit was built piece by piece. The operating sequences were recorded and integrated in the database. For every identified and tracked physical element, the appropriate virtual item needs to be displayed in its proper pose. This being said, for each database image, the next piece to be assembled is connected to it, with the help of 3D CAD modeling. The operator may visualise the virtual component, as well as its own assembly instructions. This leads to a more intuitive training, a better learning curve and a better precision at the assembly part for the coworker. The operator view is augmented with the information from the closest object.

To display the data regarding the currently tracked item, the industrial worker only needs to tap on the device's screen. This tap is converted from 2D to 3D in order to compare its position to the 3D pose of the object. Then, the requested information may be visualised in a dialog.

Several application snapshots may be seen in Fig. 17. In Fig. 17(a), it can be noticed that the assembly task is in progress. All the elements are assembled, on the breadboard the needed pieces are positioned: the Arduino, resistor, LED. The wires that connect the components are in place, as well as the connector of the power source. There is only one final task to be completed: attaching the battery. This is represented by the blue CAD model, by the means of augmentation. In Fig. 17(b) the components are arranged for the assembly task. After placing the battery, the augmented CAD model is no longer shown, so the only aspect that needs to be done is linking the battery with the battery connector. Thus, the assembly task is finished.

Static data was organised in a JSON, containing the connections between the database images, their names, the matching component to be next assembled, its pose, along with data for the currently tracked element. This JSON was then used in the application with the help of serialisation libraries and a custom model class.

4.3. Summary AR device comparison

One important aspect in the quality of the AR device is the internal-external calibration, i.e. the ability to register in time and space the real and virtual image content. According to the experiments carried out by the authors, the external calibration (e.g. with respect to a landmark) has less influence on the overlap than the initial calibration. The results of a test calibration for different AR devices is summarised in Table 4.

Beside the quality of the AR image, a number of other constrains must be taken into account. Some of the criteria are the maturity of the open source packages, cost, localisation mode, documentation, etc. A comparison between the AR technologies used (HoloLens, ARCore, Epson Moverio and Google Tango) was constructed in Table 5.

5. Conclusion

In this paper, the main challenges regarding the integration of augmented reality devices into industrial environment were presented. Beside the overview of the current trends in the state of the art of the devices for AR, practical examples were given for different experimental setups. The common parts in these setups were related to the external calibration of these devices with respect to different types of robots. Comparison of different AR approaches described in the paper helps the user to select an appropriate variant for the targeted application, in terms of the scale of the application (ranging from a few centimetres to a few meters), the calibration procedure (static, one-shot or dynamic), target libraries (ROS, Android,.NET, Vuforia) or the level of the available documentation and support.

The main findings of these experiments highlight the benefits of each approach: the HoloLens proved to be well documented, easy to be integrated even in open-source ROS industrial based environments for remote assistance applications. ARCore is still in the early release phase but is promising (running on the wide-spread Android based devices), especially for small range (under 2m) in operator training applications.

Table 5
ROS-I ready AR technologies comparison.

	HoloLens	ARCore	Google Tango	Moverio BT300
Release Date	2016, March	2018, March	2014, June	2017, January
Development Status	Stable	Stable	Discontinued	Stable
Device Type	Head-mounted	Hand-held	Hand-held	Head-mounted
Cost	$3,000	$280 +	$500	$700
Documentation	Vast	Limited	Withdrawn	Updated
Depth Sensor Type	✓	×	✓	×
Localisation mode	Motion tracking	IMU + vodo	IMU + vodo	Vodo
ROS Integration	Rosbridge	Rosjava	Rosjava	Rosjava

Although Google Tango is officially not supported anymore, for longer distance applications (up to 10m) it showed its main strength, especially for safe AGV navigation. The recent Epson HMD devices proved to be a comfortable, easily integrated devices in the remote assistance applications using Vuforia libraries. In terms of pose tracking accuracy, the devices with external depth sensors proved to be more accurate/robust in the experiments.

These experiments are easy to reproduce following the detailed description in this paper, as well using the code shared by the authors on their website.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgments

The author would like to thank to their project colleagues (Laszlo Tofalvi, Levente Bagoly, Peter Mago) for their support and involvement in realising this work. This work was supported supported by the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PN-III-P2-2.1-PTE-2019-0367, grant agreement no. 24/2020, Hungarian Research Fund grant number OTKA K 120367, ROSIN 732287 FTP, and HAS Bolyai Scholarship. The authors wish to thank the anonymous reviewers for their useful comments, which helped them to improve the presentation of the paper.

References

[1] A. Perzylo, N. Somani, S. Profanter, I. Kessler, M. Rickert, A. Knoll, Intuitive instruction of industrial robots: Semantic process descriptions for small lot production, 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2016, pp. 2293–2300.
[2] P. Milgram, H. Takemura, A. Utsumi, F. Kishino, Augmented reality: A class of displays on the reality-virtuality continuum, Proceedings Volume 2351, Telemanipulator and Telepresence Technologies, (1994), pp. 282–292.
[3] R. Palmarini, J.A. Erkoyuncu, R. Roy, H. Torabmostaedi, A systematic review of augmented reality applications in maintenance, Robot. Comput. Integr. Manuf. 49 (2018) 215–228.
[4] S. Weyer, M. Schmitt, M. Ohmer, D. Gorecky, Towards industry 4.0-standardization as the crucial challenge for highly modular, multi-vendor production systems, IFAC-PapersOnLine 48 (3) (2015) 579–584.
[5] M. Gattullo, G.W. Scurati, M. Fiorentino, A.E. Uva, F. Ferrise, M. Bordegoni, Towards augmented reality manuals for industry 4.0: amethodology, Robot. Comput. Integr. Manuf. 56 (2019) 276–286.
[6] R. Palmarini, J.A. Erkoyuncu, R. Roy, H. Torabmostaedi, A systematic review of augmented reality applications in maintenance, Robot. Comput. Integr. Manuf. 49 (2018) 215–228.
[7] D. Tatic, B. Tesic, The application of augmented reality technologies for the improvement of occupational safety in an industrial environment, Comput. Ind. 85 (2017) 1–10.
[8] D. Mizell, Boeing's wire bundle assembly project, in: F. Taylor (Ed.), Fundamentals of Wearable Computers and Augmented Reality, 2001, pp. 447–467. Ch. 14
[9] X. Wang, S. Ong, A. Nee, A comprehensive survey of ubiquitous manufacturing research, Int. J. Prod. Res. 56 (1–2) (2018) 604–628.
[10] R. Radkowski, J. Herrema, J. Oliver, Augmented reality-based manual assembly support with visual features for different degrees of difficulty, Int. J. Hum. Comput. Interact. 31 (5) (2015) 337–349.
[11] A. Blaga, L. Tamas, Augmented reality for digital manufacturing, 2018 26th Mediterranean Conference on Control and Automation (MED), IEEE, 2018, pp. 173–178.
[12] J. Kletti, Manufacturing Execution System - MES, Springer-Verlag Berlin Heidelberg, 2007.
[13] L. Tamas, M. Murar, Smart CPS: vertical integration overview and user story with a cobot, Int. J. Computer Integr. Manuf. 1 (1) (2018) 1–18.
[14] A.L. Janin, D.W. Mizell, T.P. Caudell, Calibration of head-mounted displays for augmented reality applications, IEEE Virtual Reality Annual International Symposium, VRAIS '93, Seattle, Washington, USA, September 18–22, 1993, Proceedings, (1993), pp. 246–255.
[15] Y. Liu, L. Wang, X.V. Wang, X. Xu, L. Zhang, Scheduling in cloud manufacturing: state-of-the-art and research challenges, Int. J. Prod. Res. 1 (2) (2018) 1–26.
[16] G. Chryssolouris, D. Mavrikios, N. Papakostas, D. Mourtzis, G. Michalos, K. Georgoulias, Digital manufacturing: history, perspectives, and outlook, proceedings of the institution of mechanical engineers, part b, J. Eng. Manuf. 223 (5) (2009) 451–462.
[17] P. Tsarouchi, S. Makris, G. Chryssolouris, Human-robot interaction review and challenges on task planning and programming, Int. J. Computer Integr. Manuf. 29 (8) (2016) 916–931.
[18] S. Makris, P. Karagiannis, S. Koukas, S. Matthaiakis, Augmented reality system for operator support in human-robot collaborative assembly, CIRP Ann. Manuf. Technol. 65 (2016) 61–64.
[19] T. Masood, J. Egger, Augmented reality in support of industry 4.0-implementation challenges and success factors, Robot. Comput. Integr. Manuf. 58 (2019) 181–195.
[20] P. Fraga-Lamas, T.M. Fernndez-Caram, O. Blanco-Novoa, M.A. Vilar-Montesinos, A review on industrial augmented reality systems for the industry 4.0 shipyard, IEEE Access 6 (2018) 13358–13375.
[21] R. Azuma, A survey of augmented reality presence: teleoperators and virtual environments, Surv. Augment. Reality 4 (1997) 355–385.
[22] A. Nee, S. Ong, G. Chryssolouris, D. Mourtzis, Augmented reality applications in design and manufacturing, CIRP Ann. 61 (2) (2012) 657–679.
[23] G. Michalos, S. Makris, J. Spiliotopoulos, I. Misios, P. Tsarouchi, G. Chryssolouris, Robo-partner: Seamless human-robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future, Procedia CIRP 23 (2014) 71–76. 5th CATS 2014 - CIRP Conference on Assembly Technologies and Systems
[24] Z. Bi, C. Luo, Z. Miao, B. Zhang, W. Zhang, L. Wang, Safety assurance mechanisms of collaborative robotic systems in manufacturing, Robot. Comput. Integr. Manuf. 67 (2021) 102022.
[25] G. Michalos, N. Kousi, P. Karagiannis, C. Gkournelos, K. Dimoulas, S. Koukas, K. Mparis, A. Papavasileiou, S. Makris, Seamless human robot collaborative assembly an automotive case study, Mechatronics 55 (2018) 194–211.
[26] A. Hietanen, R. Pieters, M. Lanz, J. Latokartano, J.K. Kmrinen, AR-based interaction for human-robot collaborative manufacturing, Robot. Comput. Integr. Manuf. 63 (2020) 101891.
[27] J. Sääski, T. Salonen, M. Liinasuo, J. Pakkanen, M. Vanhatalo, A. Riitahuhta, Augmented reality efficiency in manufacturing industry: a case study, DS 50: Proceedings of NordDesign 2008 Conference, Tallinn, Estonia, 21.-23.08. 2008, (2008).
[28] S. Makris, G. Pintzos, L. Rentzos, G. Chryssolouris, Assembly support using ar technology based on automatic sequence generation, CIRP Ann.-Manuf. Technol. 62 (1) (2013) 9–12.
[29] L. Tamas, Z. Kato, Targetless calibration of a lidar-perspective camera pair, ICCV 2013, BigData3DCV Workshop, (2013). http://www.cv-foundation.org/openaccess/ICCV2013_w
[30] R. Frohlich, L. Tamas, Z. Kato, Absolute pose estimation of central cameras using planar regions, IEEE Trans. Pattern Anal. Mach. Intell. 1 (1) (2019) 1–16.
[31] X. Wang, S.K. Ong, A.Y.C. Nee, A comprehensive survey of augmented reality assembly research, Adv. Manuf. 4 (2016) 1–22.
[32] Y. Lu, C. Liu, K.I.-K. Wang, H. Huang, X. Xu, Digital twin-driven smart manufacturing: connotation, reference model, applications and research issues, Robot. Comput. Integr. Manuf. 61 (2020) 101837.
[33] F. Osti, A. Ceruti, A. Liverani, G. Caligiana, Semi-automatic design for disassembly strategy planning, An augmented reality approach, Procedia Manufacturing, volume 11, (2017), pp. 1481–1488. 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy
[34] T.P. Caudell, D.W. Mizell, Augmented reality: an application of heads-up display technology to manual manufacturing processes system sciences, 1992, Proceedings of the Twenty-Fifth Hawaii International Conference on, vol. 2, IEEE, 1992, pp. 659–669.
[35] 2013 International Conference on Virtual and Augmented Reality in Education, in: J. Martín-Gutiérrez, E. Ginters (Eds.), Vol. 25 of Procedia Computer Science, Elsevier, 2013.
[36] A. Syberfeldt, O.M. Danielsson, P. Gustavsson, Augmented reality smart glasses in the smart factory, Product Evaluation Guidelines and Review of Available Products, volume 5, IEEE Access, 2017, pp. 9118–9130.
[37] ARToolKit, 2018. https://github.com/artoolkit/artoolkit5.
[38] Itseez, The openCV Reference manual, second ed., (2018). http://opencv.org/
[39] R. Horaud, F. Dornaika, Hand-eye calibration, Int. J. Rob. Res. 14 (3) (1995) 195–210.
[40] RPL, KTH Royal Institute of Technology, Camera Robot Calibration. https://github.com/kth-ros-pkg/camera_robot_calibration.
[41] B. Schmidt, L. Wang, Automatic work objects calibration via a global-local camera system, Robot. Comput. Integr. Manuf. 30 (6) (2014) 678–683.
[42] T. Foote, Turtlebot Navigation stack, (2018). http://wiki.ros.org/turtlebot_navigation
[43] E. Marder-Eppstein, Project tango, ACM SIGGRAPH 2016 Real-Time Live, ACM, 2016, p. 25.
[44] D.H. Douglas, T.K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartographica: The International Journal for Geographic Information and Geovisualization, vol. 10, (1973), pp. 112–122.
[45] Rosjava, 2018. https://github.com/rosjava/rosjava.