Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 232 (2024) 1991–2000

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on Industry 4.0 and Smart Manufacturing

Use of augmented reality for iterative robot program optimisation in robot-automated series production processes

Lukas Antonio Wulff[a,b,*], Ole Schmedemann[b], Thorsten Schüppstuhl[b]

[a]ICARUS Consulting GmbH, Friedrich-Penseler-Straße 10, 21337, Lüneburg, Germany
[b]Hamburg University of Technology, TUHH, Institute of Aircraft Production Technology, Denickestraße 17, 21073, Hamburg, Germany

Abstract

The inspection, analysis, and modification of robot programs in series production processes is a challenging recurring task in the automotive industry. Especially the holistic analysis of detected defects and the subsequent derivation and application of corrective measures to the underlying robot program is a complex task requiring extensive knowledge in multiple fields. Existing scientific literature has demonstrated that the utilisation of Augmented Reality (AR) assisted robot programming systems (ARRPS) improves programming efficiency as well as intuitiveness when compared to traditional programming methods. However, the accuracy especially of mobile AR devices limits the applicability of AR in industrial applications. We propose to leverage the iterative nature of the optimisation process common in series production and utilise the visual capabilities of the human worker to define corrective measures based on the inspected real process result. This mitigates the impact of limited tracking accuracy as the optimisation is based on the visual perception of the user and thus decoupled from the accuracy of the employed AR device. We implemented our system based on the requirements of a sealing workstation of the Mercedes-Benz Group in Germany. A conducted experiment validates the functionality and indicates that the presented strategy of iterative AR assisted robot programming enables optimisations with a similar accuracy as the Teach-In programming as well as a significantly reduced programming time. This suggests that utilising an ARRPS can be highly beneficial in the inspection, analysis, and modification of sealing applications common in automotive series production processes. As the presented iterative AR assisted robot programming is neither limited to sealing nor to the automotive sector it can be adapted for various robot-automated applications like path-welding, gluing, or painting and can extend its utility to related sectors such as aviation or the marine industry.

Keywords: industrial robot programming; augmented reality; mixed reality; automotive; simulation; sealing application

1877-0509 © 2024 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 5th International Conference on Industry 4.0 and Smart Manufacturing
10.1016/j.procs.2024.02.021

1. Introduction

The production of a car is a highly automated multilayered process involving various trades and technologies. In particular, protection against corrosion is an elementary requirement. Sealing is an important robot automated process in which corrosion vulnerable areas like metal joints are coated with a sealant material, preventing the contact with water and other media thus inhibiting corrosion. However, as the quality of each sealing seam is highly dependent on both the process parameters of the sealing process as well as the gap dimensions of the prior car body construction it is prone to errors. Due to its importance, each seam is inspected in a subsequent process. Detected defects are documented and manually corrected. The documentation is then utilised to analyse and optimise the underlying robot programs to increase the overall process stability. The defect analysis, derivation, and application of optimisations by reprogramming the involved robots is a challenging task, that requires high-level expert knowledge in multiple fields [1,2]. With the ongoing demographic change aggravating the persistent shortage of skilled workers, methods to simplify or restructure the present programming process of industrial robots in series production processes are highly sought [3].

A potential solution could involve the utilisation of Augmented Reality (AR). While various definitions exist, we view AR as a medium that seamlessly combines digital three-dimensionally registered interactive content with the basic human perception of reality to create highly immersive user interfaces [4–6]. Existing literature has already shown that AR assisted robot programming systems (ARRPS) excel in the areas of efficiency and intuitiveness, when compared to traditional programming methods like Teach-In or offline programming (OLP). However, featured ARRPS are often only tested under laboratory conditions and especially the integration into existent industrial infrastructures is not adequately explored [7,8]. In addition, especially mobile AR devices like the Microsoft HoloLens without an additional external setup of tracking hardware, often lack the accuracy needed in applications beyond simple trajectory planning and pick-and-place applications [9].

Hence, the goal of this paper is the development, implementation, and experimental evaluation of an ARRPS to improve the efficiency and intuitiveness of robot automated sealing in the automotive industry. The paper is structured as follows: Firstly, an overview of the different stages of a common car manufacturing process and the involved robot-automated processes will be given. Afterwards the sequence of defect detection, analysis and optimisation of the sealing process will be presented and specific requirements for an ARRPS will be derived. Secondly scientific literature covering ARRPS in adjacent applications will be discussed and a system design for an ARRPS will be created. Thereafter, the implementation and experimental evaluation will be described. Finally, the acquired key findings and the potential of the developed ARRPS in the field of series production processes in the automotive industry will be highlighted and necessary future steps as well as research directions will be given.

2. Iterative Robot Program Optimisation in the Automotive Industry

Due to high requirements on quality, plannability, and productivity the automotive sector pioneered in the introduction of robotic automation. Today, in addition to fully automated workstations and purely manual workstations, the proportion of collaborative robots increases [10].

Fig. 2 illustrates an exemplary linear manufacturing process as well as commonly automated tasks like part-handling, riveting, welding, sealing, and painting. Depending on the process, usually two different types of workstation setups are common. In Stop-And-Go workstations the car is mounted onto a skid, which is then moved into a workstation and fixed by a clamp. The exact position is then calculated by e.g., a camera-based 3D-Measurement and a corresponding position correction is given to the partaking industrial robots. The robots then execute their specified tasks. After the process is complete, the skid is released and automatically transported to the next workstation. Stop-And-Go workstations are commonly used in processes like welding, riveting, or sealing. An alternative, often used for painting, is a Line-Tracking workstation. This type steadily moves the skid at a low velocity and relays the current position to the employed robots. The robots, that are usually mounted on external axes, then move in coordination with the skid.

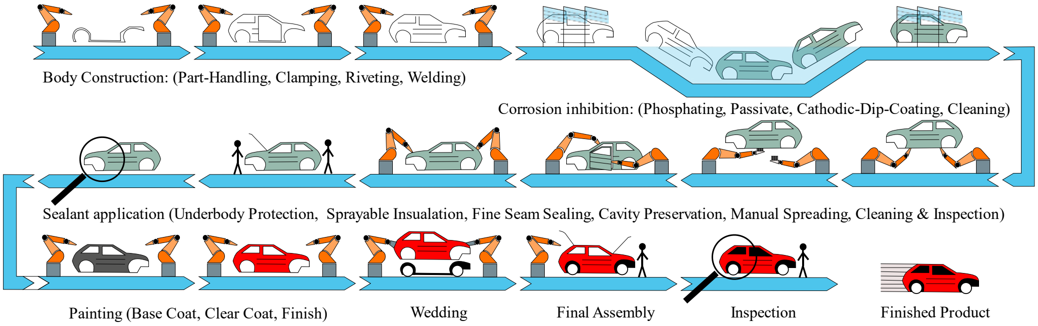

Fig. 1. Simplified overview of the different production steps during a sequential car manufacturing process. Based on [10].

An additional differentiation needs to be made between point- and path-wise applications. For point-wise applications like spotwelding, riveting or cavity-conservation, the accuracy of the pose of the tool-centre-point (TCP) and the process parameters generally define the quality of the automated process. However, path wise applications, like gluing, painting, or sealing are more complex, as the process parameters, e.g., material flow interdepend with the movement of the TCP [11]. Sealing is a crucial task for corrosion inhibition of the car body. Even though methods like cathodic-dip-coating offer a certain level of protection, vulnerable areas like welded joints are additionally protected by sealing seams [12].

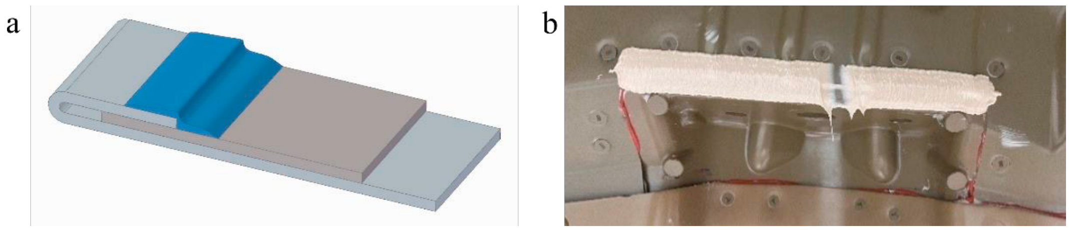

Fig. 2. Schematic presentation of a sealing seam coating two sheets (a); Slitted sealing seam in the wheelhouse of a car (b).

Fig. 2 presents a schematic sketch (a) and a real slitted PVC sealing seam (b) located in the wheelhouse of a car body. The quality of the seam depends on its position relative to the joint area and its shape. The position of the seam is determined by the relative position and orientation of the gun during application. This in turn is defined by the position and number of path points as well as movement parameters like movement-type, zone, acceleration, and jerk. The shape of the seam is influenced by gun orientation, gun distance, speed, acceleration, jerk, material flow, material temperature, and material pressure. While the partly interdependent parameters define the trajectory and thus have a major effect on the applied seam, prior production processes like body construction as well as the boundary conditions of the application (material temperature, material pressure, material viscosity, room temperature, room humidity) influence the achieved result. After all seams have been applied, the car is moved into an oven for curing. After curing each seam is inspected for defects. Detected defects are documented and then manually corrected. In a subsequent step the documentation is then analysed and depending on type, severity, and quantity of the documented defects corrective measures, e.g., necessary optimisations for the underlying robot programs are derived, and executed.

Each phase has specific challenges. During the inspection phase, it is necessary to identify any defects, extract relevant information such as defect type, severity, and associate seam. In addition, the documentation is paper-based, hence a short but precise description is required. Given that sealing typically involves a sequence of automated workstations, pinpointing the responsible robot and program causing the defective seam can be a complex task during the analysis phase. In addition, as the information density of the paper-based documentation is limited, analytical experts are often required to inspect the physical car itself to fully understand each defect. Moreover, as the cause of a defect is heterogeneous, the derivation of an optimal modification is challenging. Modifications are therefore applied iteratively and then reassessed after the next production cycle. Depending on the nature of the modification even online executed Teach-In programming might be necessary, which in turn requires a production stop and thereby causes an increase in production costs and additional stress to the responsible workers. Hence, the core challenges during modification phase are minimisation of production down time and stress of the executing worker.

In order to address the challenges inherent in the current inspection, analysis and modification process, an ARRPS needs to fulfil the following requirements:

- Increase of information density of defect documentation and simplification of defect-program allocation.
- Simplification of process analysis by visualising trajectory as well as relevant parameters.
- Simplification of optimisation by utilisation of tools to directly modify priorly visualised programs.
- Analytic tools to simulatively evaluate the effect of changes (trajectory- and cycle time calculation).
- Secure production-parallel commission of modified programs.

3. Related Work

The scientific literature presents a variety of different ARRPS. As the design space of an AR application is defined by its setup e.g., the combination of devices to realise visualisation, interaction and tracking, AR solutions can be tailored based on the specific requirements of the individual use case.

Lambrecht et al. focused on the intuitiveness of natural robot programming. They developed a programming paradigm based on human gestures, that are captured by a tablet computer. Gestures are interpreted and used to create or modify robotic paths, that are displayed as trajectories overlayed on the real three-dimensional environment. Using standardised interfaces, programs can be exchanged with connected industrial robots. A user study demonstrated that for basic pick-and-place applications the presented natural robot programming is beneficial in terms of intuitiveness as well as programming efficiency when compared to methods like Teach-In and OLP [13].

This work showcases the designability of AR applications and the potential to create highly intuitive user interfaces that open access to robot programming even to lesser skilled workers. However, even though intuitive interfaces are to be aspired the current industrially established process is centred around the procedures of shopfloor near OLP as well as online Teach-In programming and an iterative optimisation. Hence, a more coherent approach that does not alter the current industrially established process like the AR-Teach-In of Chacko et al. might be more applicable [14]. In this work a basic and cost-efficient smartphone-based AR application was used to simplify the creation of robot programs on a planar surface. The application uses iterative addition or removal of path points. With AR the end-effector, trajectory as well as the respective path points are displayed. A user study conducted in accordance with ISO 9283 compared the developed AR application to a Hand-Gesture programming paradigm [15]. In the study four different paths were created and a pointwise mean accuracy ranging from 3.65 to 4.11 mm as well as an increased programming speed, when compared to the Hand-Gesture programming was observed.

While the presented application is limited to path points on a planar surface the pointwise modification of programs as well as the visualisation of the end-effector can potentially simplify the analysis and optimisation process necessary in our aspired scope. However, the achieved accuracy does not suffice as even in rough areas like the underbody seam applications require an accuracy below one millimetre, hence an additional approach to ensure accuracy is necessary.

To engage the issue of limited accuracy in ARRPS, Ong. et al. present a system utilising an externally tracked stylus. Combined with a head-mounted-display (HMD), that visualises additional information, path points can be created and combined to a trajectory. However, unlike other approaches, the stylus is not used to directly define the trajectory of the weld path but is used to create a touch motion plan. The employed robot automatically approaches each path point and then decreases the distance between end-effector and object surface, until contact is made. The resulting pose is used to define the actual position of the surface, to then mathematically derivate the position of the seam and calculate the correct position of the corresponding weld application point. In a user study the creation of weld paths with the presented AR system has been compared to a Teach-In approach. The results show an overall accuracy below one millimetre and a reduced working time, when compared to the Teach-In method (47 to 434 seconds) [16].

The methodology presented by Ong et al. is of particular interest, as it demonstrates the partial decoupling of the weld seam's quality from the tracking accuracy of the utilised stylus. This distinguishes their approach from other scientific and industrial implementations like for example Skillreal[1], who employ powerful but expensive LIDAR sensors within specialised AR workstations to achieve submillimetre accuracy. Hence the goal of this paper is the transformation of the decoupled programming strategy presented by Ong. et al. to the specific boundary conditions of the presented series sealing process in the automotive industry. This is advantageous, as benefits, such as mobility, flexibility, and cost efficiency that AR devices with less accuracy like the Microsoft HoloLens or tablet computers offer, might be harnessed in industrial applications even though their absolute accuracy is limited.

4. Concept

In the scope of series sealing in the automotive industry, a similar degree of decoupling as demonstrated by Ong et al. may be achieved by leveraging the iterative nature of the modification process. As users inspect the physically applied seam, corrective measures can be derived relative to the real process result rather than to the digital content visible in AR. As each modification is grounded in the physical examination of the actual seam, its quality is decoupled from the accuracy of the displayed AR content. Thus, the ARRPS assumes the role of an assisting medium, that on one hand, provides additional insights into complex elements like the trajectory, robot configuration or application parameters and on the other hand provides intuitive tools to directly modify the underlying robot program based on the inspection and analysis of the real seam.

a b

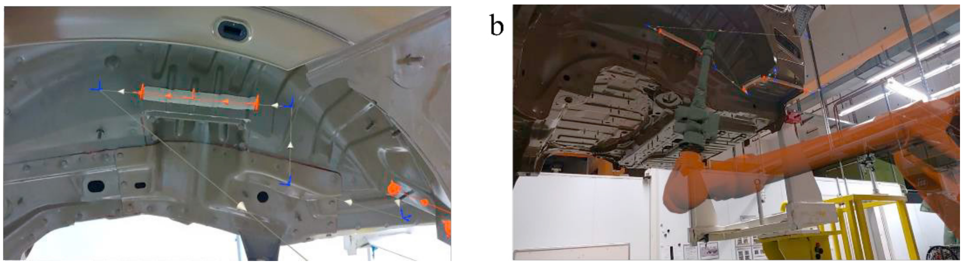

Fig. 3. AR trajectory overlaid on real seam (a); AR robot at path point (b).

Fig. 3 a presents a robot trajectory displayed in AR and overlaid on a physically applied sealing seam. The displayed content increases the understanding of the underlying robot program as position and type (blue = movement; orange = application point) of the different path points are indicated visually. Fig. 3 b displays an AR representation of the associated industrial robot when arriving at a specific path point, thus offering additional insight into the configuration. In addition, the user needs access to a sufficient user interface not only to analyse the status quo, but directly modify the underlying program. To offer further assistance simulative capabilities, e.g., movement and cycle time analysis are required to directly evaluate the effect of a potential modification. However, to enable such a workflow displayed AR content has to be accurate enough to ensure a correct allocation of the digital trajectory to the physical seam. While we assume that slight deviations are acceptable, an accuracy in a similar range as displayed in Fig. 3 a is aspired. In the context of the present iterative optimisation of sealing applications, it is noteworthy that the user can continuously assess the accuracy of the registration through visual means. If the displacement of a digital seam prevents an unambiguous allocation, the user must be able to quickly re-register the AR device. As the user continuously evaluates the quality of the registration, he assumes the role of a controlling unit, able to mitigate inaccurate tracking based on his own specific perception.

To enable this procedure, the registration and re-registration must be flexible and easy to use. In addition, measures to stabilise the displayed AR content as much as possible must be utilised. Moreover, to display and modify the underlying robot program in AR, access to the production database e.g., robot memory or program storage and a capable simulation model of each workstation is required.

[1] https://www.skillreal.com

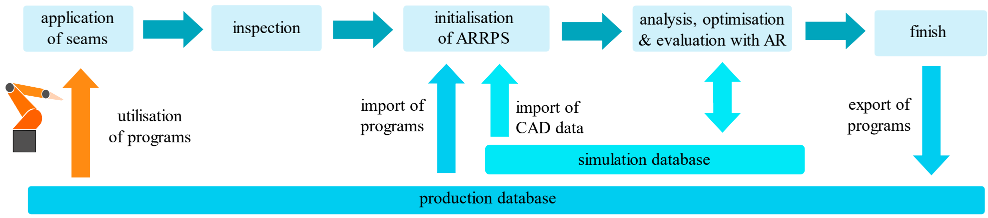

Fig. 4 Administered application sequence.

Based on the requirements the application workflow sketched in Fig. 4 was derived. After curing each seam is inspected for defects. If defects are detected, the user initiates the ARRPS and selects the corresponding car-type. The respective digital models and relevant positioning information are then imported from the connected digital database. Subsequently, the relevant programs are imported from the production database and displayed as visible trajectories superimposed directly onto the physical car body. The user then engages in a comprehensive, seam-specific examination of each defect. Depending on the particular seam being addressed at that moment, additional AR content like the associated robot is displayed. Moreover, the ARRPS offers capabilities to fully modify the underlying robot program. In addition to modifying the trajectory by moving, creating, or deleting path points additional parameters like zone, speed, acceleration, and point type are adjustable as well. Additional features include the calculation of movement simulation and cycle time measurement. After modification and evaluation in AR, the now optimised robot program can be pushed to the production database to be utilised in the next production cycle. The user then works likewise for any other defect.

5. Implementation

To gain insights on the applicability of the proposed workflow, it was implemented in an experimental setup in a sealing station of the automotive original equipment manufacturer Mercedes-Benz in Germany. As foundation for the implementation we reused our AR specific system architecture already presented in [17]. It utilises dependency injection as well as modularisation to decouple the separate AR-specific tasks (visualisation, input, and tracking) and use-case specific tasks (simulation and plant communication) into independent microservices that are orchestrated via a network-accessible server-side broker. It is based on the principles of the “clean software architecture” philosophy explained in [18]. The broker and thereby the entire application is managed by a use-case specific application running on a separate network accessible server. Each microservice operates independently from each other. Fig. 5 a outlines the general structure of the architecture and maps the subsystems used for implementation.

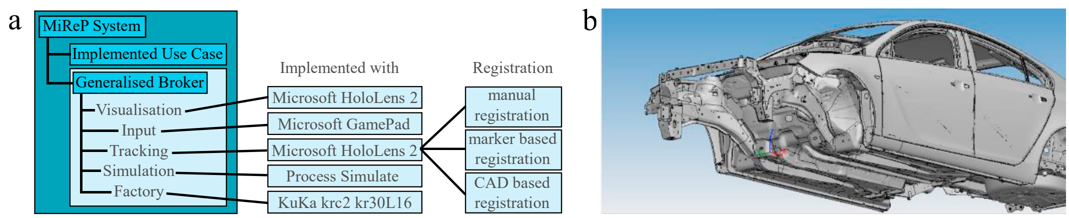

Fig. 5 Employed system architecture (a); Position of car frame relative to an exemplary car body in Process Simulate (b).[2]

The inspection and analysis of applied sealing seams require a certain level of mobility as well as the ability to handle additional tools or open elements like the bonnet, doors or hood. Hence, we employed a Microsoft HoloLens

[2] As no permission for full visualisation of the examined car body was given, a different digital model is used for visual representation.

2. This HMD, while limited in its absolute accuracy as well as field of view, fulfils these stated requirements. In addition, it has various sensors and cameras enabling the employment of SLAM and camera-based tracking. The visualisation was implemented utilising the game engine Unity[3] version 2019.4f. We chose Unity as it offers access to different AR development frameworks, is well documented and easy to use.

To increase the flexibility and enable different approaches we developed three independent strategies for registration. The core element for our registration process is the position of the work object, in our case, the car body. The position is represented by the working frame depicted in Fig. 5 b. This working frame serves as the general reference of our three registration methods and AR content is always displayed relative to this frame.

Manual Registration: The user manually aligns a displayed AR car body with its real counterpart. A Spatial Anchor is then created. A Spatial Anchor is a HoloLens-specific feature, that represents a relocatable point in the real environment. It is used to increase the stability of displayed AR content. The benefit of this variant is its quick and intuitive utilisation. The drawback is the dependence on the visual perception and positioning accuracy of the user.

Marker-based Registration: In this approach we utilise the camera of the employed HoloLens 2 and the ArUco marker tracking algorithm to automatically detect and calculate the position of known two-dimensional markers in the working field and place our AR content accordingly [19]. The user visually supervises the positioning of content and can manually pause the localisation if a proper alignment has been achieved. The main benefit of this approach is the high accuracy as well as repeatability. However, the accurate positioning of each used marker has to be ensured.

Model-based Registration: This approach uses the camera as well as the capabilities of the Vuforia SDK[4] to automatically track the position of defined CAD-elements in the real environment. It is similar to the marker-based approach but as no effort in placing the marker has to be made it is easier to use. However, the position calculation is more expensive in terms of computing power, than the marker-based approach.

The interaction is implemented with a wireless Microsoft controller. Due to its origin in the gaming industry, it is easy to use and robust enough for prolonged use. We created a simple WebSocket application that grabs the input of a controller, transmitting it to the server-side broker. The input is generalised and dynamically interpreted depending on the context of the current application.

The offline programming system Process Simulate (PS) is used as the basis for program processing and simulation. PS has access to a variety of robot programming tools and robot controllers of common manufacturers required for an accurate movement, application, and cycle time simulation. As PS offers an extensive API, we integrated it into a server-side webservice accessible via HTTP and WebSocket connection. In PS, the necessary data of the individual workstation is provided in so-called studies. Each study contains the respective car, individual robots and additional elements like external axes or tools. For our experiment we simplified the present workstation as depicted in Fig. 6.

a b

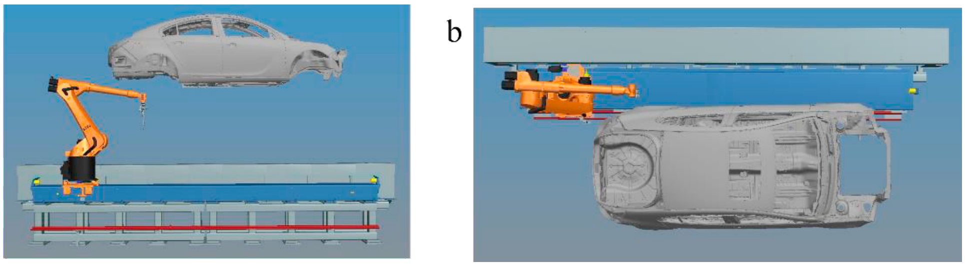

Fig. 6: Digital workstation in Process Simulate: side view (a) and top view (b).

The study consists of a single Kuka kr30L16 with a KRC2 controller that is mounted on a rail. The robot has a 3D sealing gun with three independent nozzles mounted at 0°, 45°, and 90° to the mounting frame of the end effector.

To manage import and export of programs from the active production a plant communication service, that utilises the existing plant network was developed. Depending on the manufacturer of the connected robot, different routines

[3] https://unity.com/
[4] Vuforia: https://developer.vuforia.com

can be used. Currently we have only implemented the communication with Kuka robots with the controller type KRC2 and KRC4 by using the standardised interfaces of KUKA Server.

6. Experiment

To evaluate the applicability of our implementation we conducted a proof-of-concept experiment. The task at hand was the modification of a defective sealing program utilising both our ARRPS as well as the traditional Teach-In programming. To increase realism, we used a productive robot program that applies sealing seams in the wheelhouse of the car. However, we deliberately added an error and displaced one seam, which consists of five path points, by 25 mm. Fig. 7 a compares the displaced (red) and original (green) program in PS. Fig. 7 b presents both displaced and original seam applied on the wheelhouse.

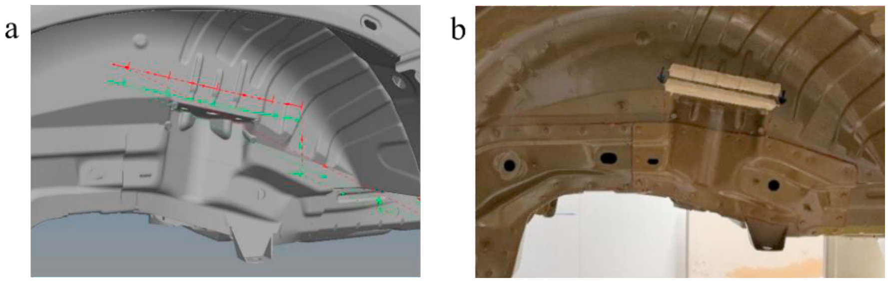

Fig. 7: Displaced program (red) and original program (green) in Process Simulate (a); Applied seams in the real car body (b).

The experiment was conducted by a professional robot teacher who is also versed in the utilisation of our ARRPS. As metrics we compared both the programming duration as well as the achieved accuracy of the real robot after modification. The employed user repeated the modification ten times for each method in an alternating order. Prior to modification the user measured the positional deviation of the applied sealant seam and calculated a displacement of 25 mm orthogonal to the application direction.

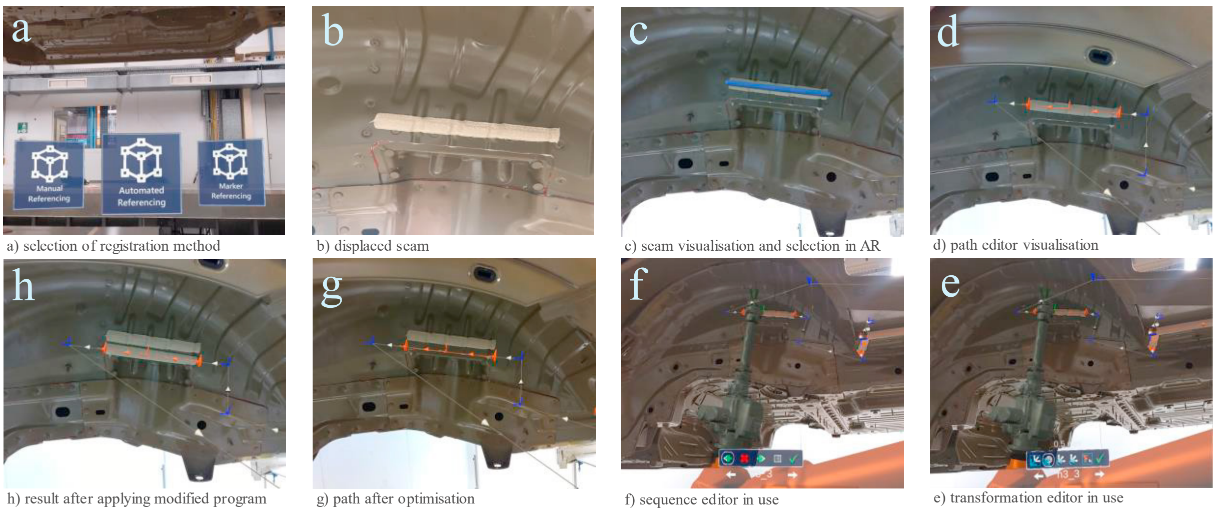

Fig. 8. AR assisted analysis and optimisation of a deficient sealing seam. Sequence is read clockwise (a-b-c-d-e-f-g-h).

Fig. 8 presents the workflow when utilising the ARRPS. Initially, the user manually selects the car type and initiates the registration. In Fig. 8 a the three different registration methods are depicted. During the experiment the model-based registration was used. After registration a Spatial Anchor is created to increase stability of the AR content. The user then proceeds to examine the car to localise defects. If a defect, as shown in Fig. 8 b, is identified the user utilises the ARRPS to assist analysis and modification. Therefore, based on the car type associated robot programs are imported from the production database and displayed as blue seams (Fig. 8 c). Due to the registration, a visual allocation of defect to digital seam and thus program is possible. The user then manually selects the digital seam to enter the AR path-editor and gain access to additional information and tools (Fig. 8 d). Parallelly the server-accessible instance of PS loads the study associated with the selected robot program and necessary components like robot, gun and external axes are imported to the AR device. The program is then displayed as a trajectory consisting of singular path points. Movement points are displayed in blue and application points are displayed in orange. Each path point is connected by a line with a cone indicating the sequence. The user can then utilise various tools grouped in editors such as the transformation editor in Fig. 8 e to modify the position and orientation of the individual path points or the sequence editor in Fig. 8 f to add or remove path points and modify underlying parameters like acceleration, speed or zone. Furthermore, the simulative capabilities of PS are utilised to calculate movement and cycle time of the modified trajectory based on the digital model that are then displayed in AR. Fig. 8 g depicts the resulting trajectory after modification. The modified program can then be pushed to the connected workstation. To simplify comparison, the real robot reruns the application with the modified program on the same car. The result is presented in Fig. 8 h.

For comparison both, position of the modified path points as well as the required programming time were taken. Similar to [14], we utilised ISO 9283, displayed in equation 1, to calculate the position accuracy of each path point PA_i.

$$PA_i = \sqrt{(\bar{x}_i - x_{c_i})^2 + (\bar{y}_i - y_{c_i})^2 + (\bar{z}_i - z_{c_i})^2} \quad (1)$$

$\bar{x}_i$, $\bar{y}_i$ and $\bar{z}_i$ are the mean coordinates for the same point optimisation repeated ten times. x_{c_i}, y_{c_i}, and z_{c_i} are the coordinates of the desired pose. We then averaged the accuracy over all path points and experiments. The mean programming time, mean accuracy and respective standard deviations are displayed in Table 1.

Table 1: Mean programming time, mean accuracy and respective standard deviations calculated with ISO 9283.

Method	Mean Programming time (s)	Standard Deviation (s)	Mean Accuracy (mm)	Standard Deviation (mm)
AR	24.6	4.5	0.12	0.32
Teach-In	92.9	14.5	0.11	0.09

Based on the result of Chacko et al. [14], we expected a significantly reduced working time as well as a significantly decreased accuracy, when comparing our AR assisted robot programming with the Teach-In method. Hence, two independent sample t-test with an alpha value of 0.05 were conducted. In case of programming time, a Levene’s test showed the homogeneity of variances and thus an independent samples t-test with homogeneous variances was conducted. In case of programming accuracy, a Levene’s test showed the inhomogeneity of the variances and thus an independent samples t-test with inhomogeneous variances was conducted. The results of the t-tests indicated a significant difference in programming time and a non-significant difference in accuracy.

7. Discussion, Conclusion and Outlook

The presented ARRPS can be utilised to assist the inspection, analysis, and modification of an automotive sealing application, thus demonstrating the applicability of ARRPS’s in the field of series industrial robot programming. The results of the conducted experiment, despite being limited by the number of participants, indicate that our iterative AR assisted robot programming enables relative modifications in a similar tolerance field as the traditional Teach-In programming while using significantly less programming time. Hence, the discussed strategy of leveraging the iterative nature of the optimisation enables the use of less accurate mobile AR devices in industrial robot programming. Thus, increasing the range of available devices and simplifying the introduction of AR to industrial applications. In addition, as our general application design is neither limited to sealing nor to the automotive sector, it can be adapted for various robot-automated applications like path-welding, gluing, or painting and can extend its utility to related sectors such as aviation or the marine industry.

Although confined in scope, the conducted experiment serves as a proof of concept paving the way for subsequent studies. In the future we will extend our research by conducting additional studies with a larger cast of participants as well as more diverse categories of defects and modification tasks.

Acknowledgements

The car body as well as working environment in which the experiment was conducted, was provided by the Mercedes-Benz Group in Bremen Germany. We would like to thank the responsible representatives for the chance to evaluate our work in an industrial setting.

References

[1] Z. Pan, J. Polden, N. Larkin, S. van Duin, J. Norrish, Recent progress on programming methods for industrial robots, Robotics and Computer-Integrated Manufacturing 28 (2012) 87–94.

[2] A. Mark, R. Bohlin, D. Segerdahl, F. Edelvik, J.S. Carlson, Optimisation of robotised sealing stations in paint shops by process simulation and automatic path planning, IJMR 9 (2014) 4.

[3] I. Hamm, H. Seitz, M. Werding, Demographic Change in Germany, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[4] R.T. Azuma, A Survey of Augmented Reality, Presence: Teleoperators & Virtual Environments 6 (1997) 355–385.

[5] M. P., K. F., A Taxonomy of Mixed Reality Visual Displays, IEICE Transactions on Information and Systems 77 (1994) 1321–1329.

[6] P.A. Rauschnabel, R. Felix, C. Hinsch, H. Shahab, F. Alt, What is XR? Towards a Framework for Augmented and Virtual Reality, Computers in Human Behavior 133 (2022) 107289.

[7] Z. Makhataeva, H. Varol, Augmented Reality for Robotics: A Review, Robotics 9 (2020) 21.

[8] J. Egger, T. Masood, Augmented reality in support of intelligent manufacturing – A systematic literature review, Computers & Industrial Engineering 140 (2020) 106195.

[9] T. Frantz, B. Jansen, J. Duerinck, J. Vandemeulebroucke, Augmenting Microsoft's HoloLens with vuforia tracking for neuronavigation, Healthc. Technol. Lett. 5 (2018) 221–225.

[10] H.-H. Braess, U. Seiffert, Vieweg Handbuch Kraftfahrzeugtechnik, Springer Fachmedien Wiesbaden, Wiesbaden, 2013.

[11] 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), IEEE, 2020.

[12] N. Akafuah, S. Poozesh, A. Salaimeh, G. Patrick, K. Lawler, K. Saito, Evolution of the Automotive Body Coating Process—A Review, Coatings 6 (2016) 24.

[13] J. Lambrecht, J. Krüger, Spatial Programming for Industrial Robots: Efficient, Effective and User-Optimised through Natural Communication and Augmented Reality, AMR 1018 (2014) 39–46.

[14] S.M. Chacko, A. Granado, V. Kapila, An Augmented Reality Framework for Robotic Tool-path Teaching, Procedia CIRP 93 (2020) 1218–1223.

[15] Industrieroboter - Leistungskenngrößen und zugehörige Prüfmethoden, ISO 9283:1998-04.

[16] S.K. Ong, A.Y.C. Nee, A.W.W. Yew, N.K. Thanigaivel, AR-assisted robot welding programming, Adv. Manuf. 8 (2020) 40–48.

[17] L.A. Wulff, M. Brand, J.P. Schulz, T. Schüppstuhl, Feasibility of Augmented Reality in the Scope of Commission of Industrial Robot Plants, in: A.-L. Andersen, R. Andersen, T.D. Brunoe, M.S.S. Larsen, K. Nielsen, A. Napoleone, S. Kjeldgaard (Eds.), Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems, Springer International Publishing, Cham, 2022, pp. 207–215.

[18] R.C. Martin, K. Henney, Clean Architecture: A craftsman's guide to software structure and design, Prentice Hall, Boston, Columbus, Indianapolis, New York, San Francisco, Amsterdam, Cape Town, Dubai, London, Madrid, Milan, Munich, Paris, Montreal, Toronto, Delhi, Mexico City, São Paulo, Sydney, Hong Kong, Seoul, Singapore, Taipei, Tokyo, 2018.

[19] M. Brand, L.A. Wulff, Y. Hamdani, T. Schüppstuhl, Accuracy of Marker Tracking on an Optical See-Through Head Mounted Display, in: T. Schüppstuhl, K. Tracht, D. Henrich (Eds.), Annals of Scientific Society for Assembly, Handling and Industrial Robotics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2020, pp. 21–31.