# SUPPORTING MANUAL CONTOUR AND SURFACE PROCESSING WITH THE HELP OF AUGMENTED REALITY

**MATTHIAS MÜHLBAUER[1], ALEXANDER AUER[1], HUBERT WÜRSCHINGER[1], NICO HANENKAMP[1]**

[1]University Erlangen-Nuremberg, Institute of Resource and Energy Efficient Production Systems, Fürth, Germany

**DOI: 10.17973/MMSJ.2023_12_2023091**

**Matthias.Muehlbauer@fau.de**

When manually processing complex contours or surfaces, there is a risk of incomplete or unnecessary multiple processing. The support of such processes with an augmented reality (AR) application will be presented in this work using the example of manual deburring.

Firstly, the positions of the component and tool are determined based on camera recordings. For this purpose, approaches for 6 Degrees of Freedom (DoF) pose estimation will be utilized. Secondly, the correct and complete processing is concluded by further analyzing the extracted data. For example, the distance of the objects to each other. In the third step, the processing result is visualized to aid the worker. Finally, the system is validated through a study. The results show that with the help of image targets a robust recognition of the objects at a normal execution speed and thus process support can be realized.

**KEYWORDS**

Augmented reality, assistance system, object detection, manual processes, contour and surface processing, deburring

## 1 INTRODUCTION AND MOTIVATION

Despite the trend towards automation, manual processes are still a significant factor in factory operations [Kellner et al. 2020]. The reasons for this are lower initial costs, high flexibility of manual processes, and the uneconomical or difficult automation of processes. Particularly, in the case of complex processes with different workstations, execution by humans is common and is to be expected in the future [Ittermann et al. 2015; Kinkel et al. 2007]. The manual execution, however, can lead to accidents, errors, or unstable process times due to increased complexity, fatigue, time pressure and different qualification levels [Henke 2015; Schenk 2015]

The manual processing of contours or surfaces represents a subarea of manual processes. Examples of such processes are cleaning, coating, welding or deburring. For this type of process, specific risks exist regarding an incomplete or unnecessarily repeated processing of sub-areas.

In this paper, a camera-based method to reduce these risks is proposed. Camera-based approaches offer the possibility to monitor the process without technical intervention [Zhang et al. 2019] and to extract a wide range of process information [zur Jacobsmuhlen et al. 2013].

With the help of recorded image data and its subsequent analysis, the extraction of the position of the relevant objects in the image is made possible. By evaluating the movement of the objects, the correct and complete processing can be concluded. A visualization serves as a support for the employee.

In the following, the relevant literature on this topic and the research gap are depicted. The method is introduced in the next section, and the results of the preliminary test are presented. The validation of the method is then carried out using the use case of manual deburring. In the last section, the results are discussed, and a summary is given.

## 2 STATE OF RESEARCH AND RESEARCH GAP

Following, an overview of different approaches for the recognition and analysis of manual processes based on image data is given. Further, the possibilities of detecting objects in an image with a focus on 6DoF pose estimation are presented. Current approaches to support deburring processes are explained in the third section. Finally, the research gap is outlined.

### *2.1 Process progress recognition based on image data*

Different use cases and approaches for camera-based detection of process actions or states have been published. Their implementation can be divided into three steps: Data acquisition, feature extraction, and recognition or classification of the action or state.

However, there are significant differences in the procedures used for feature extraction and state or action detection, see Figure 1. Either a manual, self-programmed set of rules or a machine learning (ML) approach is used to perform these tasks. Firstly, approaches that use a set of rules for this purpose are discussed. Secondly, publications that use ML are reviewed.

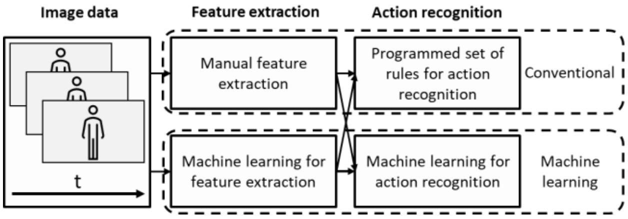


**Figure 1.** Approaches to action recognition based on image data

Faccio et al. monitor an assembly process, based on the detection of hand movements, which must take defined positions in space [Faccio et al. 2019]. The correct positioning of the component is not verified. The proposed procedure by Manuri et al. includes a comparison of the shape of the assembled objects with a defined state [Manuri et al. 2019]. This allows the verification of the correct positioning. An approach to determine the speed of motion in a manual welding process was developed by Digiacomo et al. [Digiacomo et al. 2020]. By processing the data from a vision system and an inertial unit, the speed of the weld pool is inferred. The fact that the weld pool is a light-emitting target is utilized for this purpose.

A two-step approach based on ML was developed by Roth et al. In the first step, the joint points of the worker are detected, which are subsequently used as features for the action classification [Roth et al. 2020]. Similar approaches have been presented by Yang et al. [Yang et al. 2020] and Fukuda et al. [Fukuda et al. 2020]. Yang et al. also extract the joint positions first. These are subsequently used for action classification with the help of a graph convolutional network. In addition to joint points, Fukuda et al. extract the position of relevant objects through an object detection algorithm. In the second step, the obtained position data are used as features in a Hidden Markov Model for action classification [Fukuda et al. 2020].

If ML is used for action or state recognition, there are inherent disadvantages. The three most significant are: Firstly, different states or actions must be learned separately. Secondly, the decision process of the model is difficult to comprehend. Thirdly, the examination of complex manual movements or questions regarding the processing of a contour or surface can hardly be covered.

Therefore, in this paper, a two-step approach is selected to detect manual actions. In the first step, hand-crafted features or ML approaches are used to detect the relevant objects in the image. In the second step, a defined set of rules is applied to evaluate the action based on the detected objects.

### *2.2 6DoF pose estimation*

Different approaches are available for the detection of a previously known object in an image. Regarding the extractable information concerning the detected object, the most common approaches are image classification, object detection, instance segmentation, and 6DoF pose estimation [Rudorfer 2021].

6DoF pose estimation provides a fully comprehensive description of the position of an object in the image. For the implementation, ML methods as well as methods based on feature matching can be applied [Sahin et al. 2020]. For approaches based on ML, a further distinction can be made between classification, regression, or combined approaches.

In feature matching, a differentiation can be made between methods based on template matching and point-pair feature matching. Here, evaluation criteria for features are applied to input images, which are stored in a hash table (point-pair feature matching) or based on special descriptors (template matching) [Sahin et al. 2020]. If sufficient features are detected, the position of an object can be determined.

Bin picking and AR-based assistance systems can be considered typical industrial applications, which require accurate position detection of an object. 6DoF pose estimation methods are used accordingly in these areas and are a subject of current research [Cui et al. 2022; Yan et al. 2020].

### *2.3 Deburring*

The developed method is validated through the use case of manual deburring. A burr is a structure created on the workpiece surface while manufacturing a workpiece, which protrudes beyond the intended and existing workpiece surface [Beier 2015]. The reasons for the necessity of removing the burr can be divided into functional (e.g. disturbance due to detaching burrs in the device), ergonomic (e.g. risk of injury) and aesthetic (e.g. validity benefit) (Schäfer und (Keine Angabe) 1975)]. Due to the automation effort, this is a typical task that is often performed manually for very small and small series.

However, the research in the field of deburring is currently focusing on the automation of this production step. The publications mainly deal with the automatic detection of contours requiring deburring as well as the following automated execution, e.g. with the help of a robot.

Song and Song as well as Tellaeche and Arana developed systems in which deburring contours are provided by matching CAD models [Song and Song 2013, Tellaeche and Arana 2016]. In the system of Song and Song, the contact force of the tool is also measured to compensate position deviations. Princely and Selvaraj presented an approach where deburring contours are determined by 2D surface detection followed by edge detection. In this system, each detected edge is defined as a contour to be machined. A procedure to localize the profile of components was suggested by Ferrari et. al. [Ferrari et al. 2015]. The deburring process is then performed by a robot. Stan et al. developed a digital twin for a robotic deburring work cell. After automated deburring, an optical inspection of the component is performed. Subsequently, only areas where the burr could not be satisfactorily removed are machined again [Stan et al. 2022].

The use of a robotic system is usually a production-increasing measure, which, however, is associated with high investment costs and frequently entails a reduction in flexibility in contrast to manual execution.

### *2.4 Research gap*

After a thorough literature search, no flexible and easily adaptable camera-based assistance system could be identified that supports a manual contour or surface processing of a workpiece by an exact spatial position analysis (6DoF) of a hand-held tool. Existing solutions often use multiple sensor systems or represent special solutions for specific use cases. Thus, there is a need to support such processes as well as to develop a corresponding method and evaluate its usability and robustness.

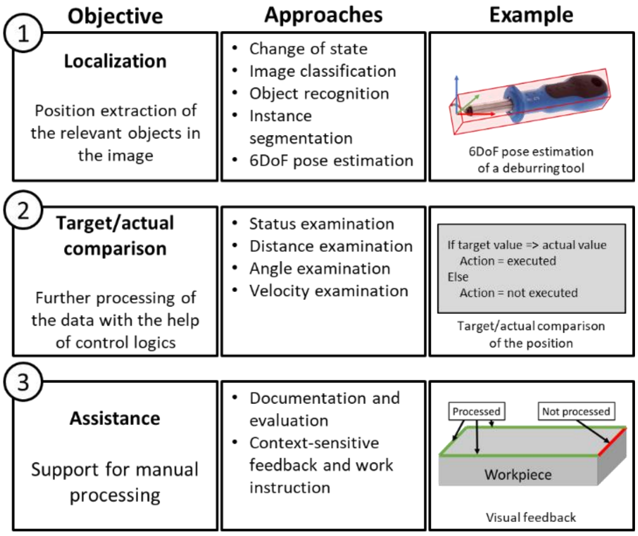
## 3 DEVELOPMENT OF A METHOD FOR SUPPORTING MANUAL CONTOUR AND SURFACE PROCESSING

The method to be developed aims to support manual contour and surface processing. Specifically, the worker is to be provided with feedback on areas that have already been processed as well as areas that may not have been completely processed.

However, it is often difficult to directly measure the completeness or quality of the process execution. If the initial state of the object differs only minimally from the target state from an optical point of view, as in the case of a deburring process, for example, this can only be monitored with correspondingly precise sensor technology. If no changes are made to the object, as in a manual haptic inspection, the process execution can only be monitored indirectly.

Therefore, in the following, a camera-based method which indirectly assesses the process execution is presented. The method is based on the understanding that manual processes can be considered as the manipulation of objects by hand to achieve a defined state [Mühlbauer et al. 2022]. During a manual contour and surface processing process, the hand or a guided tool is moved over the workpiece with the aim of inspecting or processing it.

By tracking the movement of the hand or tool in relation to the object to be machined and a subsequent comparison with a target process, it is thus possible to indirectly assess the process execution. For the realization of an assistance solution based on this foundation, a three-step approach is proposed, see Figure 2.



| | Objective | Approaches | Example |
|---|---|---|---|
| 1 | **Localization**<br>Position extraction of the relevant objects in the image | • Change of state<br>• Image classification<br>• Object recognition<br>• Instance segmentation<br>• 6DoF pose estimation | 6DoF pose estimation of a deburring tool |
| 2 | **Target/actual comparison**<br>Further processing of the data with the help of control logics | • Status examination<br>• Distance examination<br>• Angle examination<br>• Velocity examination | If target value => actual value<br>Action = executed<br>Else<br>Action = not executed<br>Target/actual comparison of the position |
| 3 | **Assistance**<br>Support for manual processing | • Documentation and evaluation<br>• Context-sensitive feedback and work instruction | Visual feedback |

**Figure 2.** A three-step approach to support manual contour and surface processing

In the first step, it is necessary to detect or localize the relevant objects in the image. Various approaches are available for this purpose. For the indirect evaluation of the execution of a contour or surface processing, the most exact localization of the objects in space should be attempted. Methods from the field of 6DoF pose estimation allow an exact description of the spatial position of objects and are, therefore, chosen for the detection of the objects within the proposed method.

In the second step, the information received regarding the object positions can be used for a target/actual comparison. For the further processing of the object positions and the realization of the monitoring of complex manual processes, control logics are developed.

In the third step, the results of the target/actual comparison can be used to support manual processing. The focus is on context-sensitive and real-time capable feedback on the work progress. Regarding the type of feedback, visual feedback is selected. Visual feedback is predestined for comprehensible feedback for contour and surface processing, since, for example, areas that have not yet been processed can be easily visualized. The implementation is realized in the context of an AR application.

In the following, especially the first two steps of the method, the detection of the objects in the image as well as the possibilities of a target/actual comparison based on control logics, are presented in more detail. Further the technical realization of the systems is presented. Finally, the limitations of the system, with a focus on 6DoF pose estimation and position control, are determined and discussed based on preliminary tests.

### 3.1 Realisation of the 6DoF pose estimation

For the 6DoF pose estimation, feature matching models were used. Already established models prove to be computationally efficient and stable in detection. Their architecture makes them especially suitable for the design of real-time applications. In addition, there is no requirement for the time-consuming creation of training data sets, as in the case of models with neural networks that aim for 6DoF pose estimation.

A 6DoF pose estimation in form of classes is not possible with feature matching models. However, according to the current state of the technology, this is also not yet fully developed with other 6DoF pose estimation models and is therefore only possible within a limited framework. Thus, this apparent disadvantage for the feature matching models is currently negligible from this perspective.

The 6DoF pose estimation software used in this work and the corresponding implementation of the control logics were realised in Unity 3D. This is a runtime and development environment for 3D applications and is suitable for the realisation of AR projects due to the possibility of creating application-oriented spatial scenes. Because of its open software architecture, this software also offers the possibility of creating and using specific extensions.

Vuforia is an extension for Unity 3D that enables camera-based 6DoF pose estimation. With Vuforia, the complete spatial position of an object is determined via reference points of a given target according to the principle of feature matching. The targets must be prepared for use in a separate processing procedure. There are different target variants, whereby only image and model targets are used within this paper.

Image targets are 2D images that Vuforia can recognise and track by comparing stored features. In Vuforia, a feature is a sharp, peaked, or angular detail in an image. Three examples of image targets with the corresponding marked features that were used in this work are shown in Figure 3.

The second type of targets considered in this work are model targets. With these, the detection of objects is achieved by matching shape features that are derived from 3D CAD models or 3D scans. In this work, the models are created via the CAD system of Siemens NX and converted into an appropriately usable file format using MeshLab. Figure 4 shows an example model from a defined perspective with a corresponding feature representation.

**Figure 3.** Image targets with corresponding features in yellow

Vuforia uses the detection process to determine the positions and orientations of the central coordinate systems of the individual targets within a scene. These are the fixed points between the real world and a created application scene that contains the corresponding control logics.

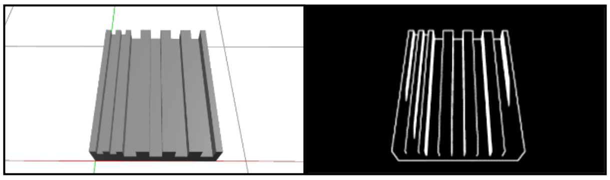


**Figure 4.** CAD model (l.) and corresponding perspective features (r.)

### 3.2 Development of control logics

To further enhance the information content of object recognition and to create a greater benefit from it, it can be completed with control logics. Control logics are automated tests that run during a process to evaluate states. Based on this, it can be determined if an action meets expectations, if it is incorrect or has not yet been executed. Since these logics can be used to evaluate status queries and relations between objects (especially in the case of a 6DoF pose estimation), they can provide additional information for the execution of processes.

Control logics are based on parameters such as detection status, position, angle, and velocity. From these, for example, the behaviour of a tool relative to a component can be determined to draw conclusions about process execution. The individual inspection parameters use the following characteristics:

**Status control.** For this purpose, Boolean expressions (true/false) are used, which are linked via logical operators. The detection status of an object in a recording is stored as a Boolean expression.

**Position control.** For the description of these logics, mathematical principles from (vector) geometry are the basis. With position-based analysis, it is possible, for example, to determine whether a tool is located at the correct processing position on the workpiece.

**Angle control.** Angle-based control logics also use position parameters as basic elements. A central vertex and two ordered points or (direction) vectors are required to describe an angle.

**Velocity control.** In addition to position components, time frames are used here. A control logic measures the change in position of a detected object for a defined time interval.

In the context of this work, control logics in the form of position controls were used. The position values for an object determined via the object detection process are stored in vectors that are based on a shared central coordinate system (usually the camera position is the origin). To reduce the complexity of a position-related control logic description it is recommended to define additional (auxiliary) coordinate systems. A vector $\boldsymbol{x}$ can be transformed via a rotation matrix $\boldsymbol{A}$ and a translation vector $\boldsymbol{b}$ into a vector $\boldsymbol{x}'$ in a corresponding coordinate system:

$$\vec{x'} = A\vec{x} + \vec{b}$$

The creation of rotation matrices is usually a time-consuming computational process. In the case of a 6DoF pose estimation, the object translations as well as the object rotations are determined in a central coordinate system, whereby corresponding rotation matrices and translation vectors for coordinate systems on detected objects are already available. In many cases, rotations are specified in Euler angles or as (unit) quaternions.

Once the coordinate systems for an implementation are defined, position control logics can be described in them. In this work, vector equations were used for discrete point descriptions. The corresponding discrete points can be defined with sufficient accuracy for real-world applications. However, with a discrete point description, additional metrics for distance determinations need to be used to generate corresponding logic trigger areas. Furthermore, due to the accuracy fluctuations in the object detection process, threshold values must be used as tolerance levels so that the corresponding control logics are triggered properly. An experimental determination of these values for a given system architecture is necessary. The vector equations used in this work as well as distance equations are listed in the following:

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \in \mathbb{R}^3 \qquad \text{(single position point)}$$

$$\vec{x} = \lambda\vec{u} + \vec{x}_0 \qquad \text{(points on a straight line)}$$

$$\vec{x} = \lambda\vec{u} + \mu\vec{v} + \vec{x}_0 \qquad \text{(points on a plane)}$$

$$\vec{x} = \lambda\vec{u} + \mu\vec{v} + \rho\vec{w} + \vec{x}_0 \qquad \text{(points in (vector) space)}$$

$$d(\vec{x},\vec{y}) = \|\vec{x} - \vec{y}\|_2 = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$
(Euclidean distance)

(with scalar $\lambda, \mu, \rho \in \mathbb{R}$ and $\vec{u}, \vec{v}, \vec{w} \in \mathbb{R}^3$ linear independent direction vectors and location vector $\vec{x}_0$)

Using the formulas described, control points which are referenced to coordinates of an object detection can be placed at defined distances (e.g. as a grid layout via the vector equation of a plane (Figure 5)) in a working plane, for example, for surface processing. As soon as a trigger point, which is located on a tool and can be defined in an analogous way to a control point, falls below a defined threshold value to a control point on the workpiece, the logic on the workpiece triggers and as a result, a visual representation dependent on it can change (e.g. defined area around a test point changes colour).

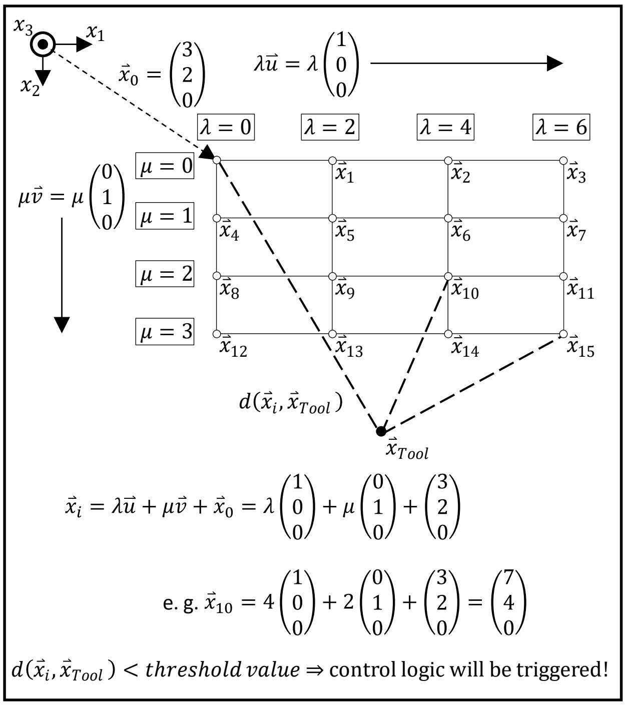


**Figure 5.** Mathematical concept for position control

### 3.3 *Technical realization of the support system*

The technical realization of a system and its configuration to support contour or surface processing requires the interaction of three elements. These are the software, the hardware and the manual process itself, see Figure 6.

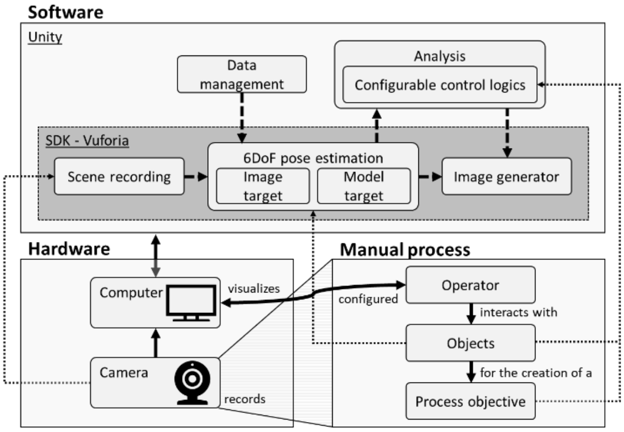


**Figure 6.** Interaction between software, hardware and manual process

Central tasks of the software are the 6DoF pose estimation via image or model targets as well as the process analysis with the help of the control logics. The implementation of the 6DoF pose estimation was already described in section 3.1. The control logics were implemented through programmed scripts in Unity 3D. For monitoring different process scenarios (e.g. different radii or contour lengths), there is the possibility of a simple adaptation of the scripts via defined configuration variables to a limited extent. The program flow chart for the analysis of a contour or surface processing with a tool is shown in Figure 7.

The required hardware consists of a laptop and a camera. As laptop, a Lenovo Yoga 530 with an Intel Core i5-8250U processor and 8GB RAM is used, which can be considered an average configuration of a mobile personal computer. This configuration was chosen because it was aimed from the outset that the developed system is functional and real-time capable without special hardware. The laptop is utilized for the configuration and execution of the software as well as for the information output to the worker. An Intel RealSense D435i camera serves to record the manual process [intel 2023]. The camera is flexibly connected to the computer with a USB cable. Thus, it is possible to mount it on a tripod or to wear it as a head-mounted camera, which allows recording the process from an egocentric perspective.

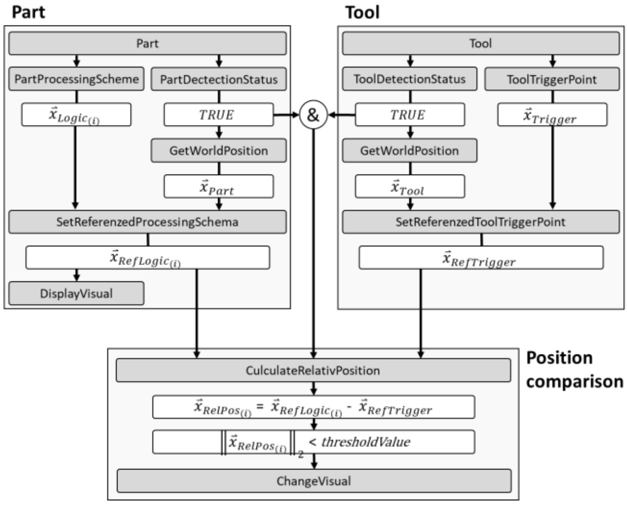


**Figure 7.** Program flow chart

The focus of the analysis is the manual process execution. Based on the process objective and the involved objects, the control logics are selected and configured. Either image targets are attached to the involved objects or they are represented by model targets. This enables their recognition by 6DoF pose estimation software.

For the following test series as well as for the validation of the use case the described software and hardware components were used. The influence of the system configuration was not investigated.

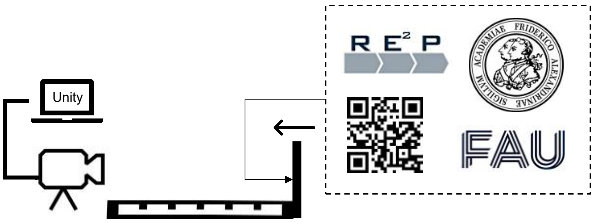
### 3.4 Investigation of the limits and precision of the system

To generate an understanding of the precision of the 6DoF pose estimation, several preliminary series of tests were performed. The focus was on the detection of image targets, as these are expected to provide the most precise and stable detections. On the one hand, an experiment on target detection was realised to provide information about the necessary size of a target for an application. On the other hand, the detection-dependent precision of control logics was analysed, e.g. the precision with which the object position and thus the tolerance range in which control logics operate were determined.

In the first experiment, the size of a target required for successful detection was determined. For this purpose, the dependency of the detection distance from the area of a target as well as the feature contents contained within it were evaluated. Four different image targets were used and each of them was also varied in size.

The corresponding experimental setup can be seen in Figure 8. This includes the respective image targets, which are attached to a plane, as well as a stationary-mounted camera. A folding ruler was used to determine the distance. The object detection software was started while the target was at a maximum distance (200cm) from the camera. From this point, the target was only moved in the depth direction towards the camera and as soon as a detection of the target occurred the distance was measured. For each size of the target, three detections were made under constant conditions.

The evaluation of the measurement series can be seen in Figure 9. It shows that the targets require a minimum size for detection. At close range, however, there is a detection advantage for targets whose features are sufficiently distinctive even in smaller areas. The graph shows a quadratic relationship between the target area and the detection distance. Thus, to achieve a double detection distance, a four times bigger target area is required, regardless of the respective feature content.



Measured variable: Distance between target and camera

**Figure 8.** Experimental setup for measuring the target detection distance

To investigate the precision of control logics based on 6DoF pose estimation, test series were designed that work with logics that are dependent on a position-based analysis. For a created position logic, the triggering distance was measured as a dependency of the target distance and the recording angle.

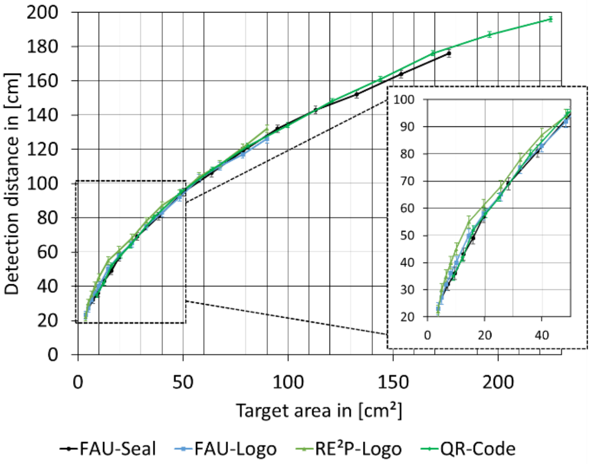
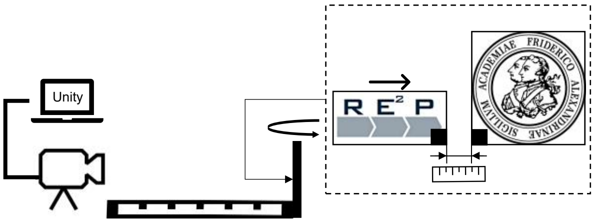


**Figure 9.** Dependency between detection distance and target area

The experimental setup is shown in Figure 10. Two targets lie in a plane, positioned at a fixed distance from the camera. The two targets were moved planar towards each other until the implemented control logic was triggered. The logic triggers are located at the corners of the targets (symbolised as black rectangles in Figure 10, activation on edge contact). When the control logic was triggered, the distance between the two corners was measured.



Measured variable: Distance between the corners of the targets

**Figure 10.** Experimental setup for measuring the precision of control logics

In the first series of experiments, the distance between the targets and the recording camera was varied up to a maximum of 55cm (maximum detection distance for the target sizes used: RE²P-Logo 10cm², FAU-Seal 19.6cm²) and the camera remained fixed orthogonally to the plane of the targets (recording angle: 0°). One of the targets was fixated so that it was in the same position for every camera distance. Only the second target was moved to trigger the control logic. The triggering distance was determined three times per distance level.

The evaluation of the measured values resulted in the graph shown in Figure 11. This shows that for the targets used the triggering distance of the control logic increases at distances of less than 30cm and more than 45cm. Thus, their precision decreases and an increased probability of errors is the consequence. The reason for this is that the targets are detected less precisely in these ranges, which is coherent with the results of the experiments on target detection (approaching the target detection limit). An optimal detection range lies between 30cm and 45cm. Within this range, the used control logic works in a stable and precise way, which results in a deviation of a few millimetres.

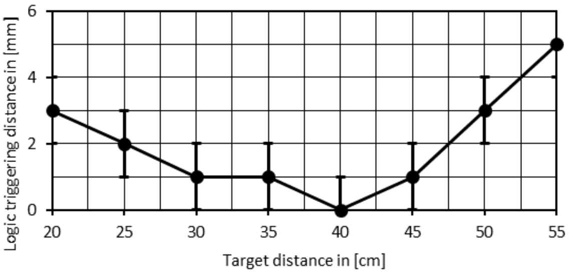


**Figure 11.** Dependency between logic triggering distance and target distance

In the second series of experiments, the angle dependency of the implemented control logic was tested. The distance between the targets and the camera was fixed at an optimised detection distance (here 40cm) and the plane on which the targets are located was set to fixed rotation angles relative to the camera. For each angular position, the logic trigger distance was measured in the same way as in the first series of experiments.

The evaluation of the measurement results can be seen in Figure 12. It shows that at a low recording angle the implemented control logic is afflicted with a deviation in the low millimetre range, which indicates a high-quality detection. As the recording angle increases, the triggering distance also increases. Thus, for angles up to 35°, deviations of a maximum of 7mm occur. Above 35°, the triggering distance increases over-proportionally and with it the associated control logic error. Above this value, a high-quality detection of the used targets is no longer possible.

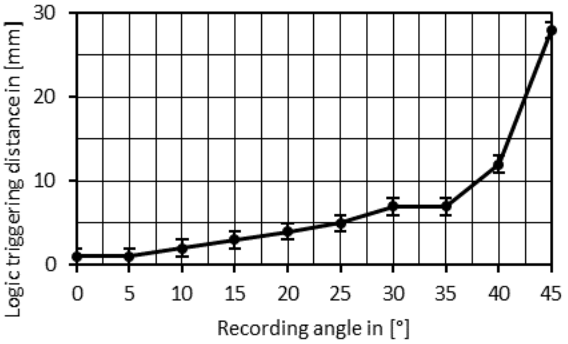


**Figure 12.** Dependency between logic triggering distance and recording angle

From the results of the previous series of experiments, it is possible to estimate the required target sizes (see Figure 9) and the tolerance range of control logics for practical applications. However, it should be noted that numerous other factors influence an object's detection. These include, for example, aggravating influences such as reflections, occlusions, and symmetries. These influences are not quantified in detail within the scope of this work.

## 4 VALIDATION BASED ON THE USE CASE OF MANUAL DEBURRING

The developed system is validated with the use case of manual deburring. During execution, a deburring tool must be moved along relevant contours. By tracking and referencing the objects involved (workpiece and deburring tool) and with the help of (position-based) control logics, it should be indirectly concluded if the processing is complete. Within the use case, the triggering behaviour of the control logics is analysed under different conditions to evaluate the practical performance of the system. The setup and layout of the experiment, the experimental procedure, and the results are described in the following sections.

### *4.1 Description of the setup*

The structure of the workstation used is as follows: a camera directed at the workspace records the movements of the tool and the workpiece. The analysis of the working process takes place during the operation. The results of the analysis are visually displayed to the operator in real time. An individual adjustment of the workplace is possible at specific points. For example, the recording device can be attached to the workstation in a static position or worn as a dynamic head-mounted camera. There are also various options for visualisation. For instance, the display of the laptop on which the software runs can be used. A separate monitor or AR glasses can also be considered for providing information.

For the implementation realised, a head-mounted camera is used, as this confronts the system with the dynamics of a scene and also represents a cost-effective variant compared to AR glasses. A laptop serves as an execution platform for the software and as an information display. A schematic layout is shown in Figure 13.

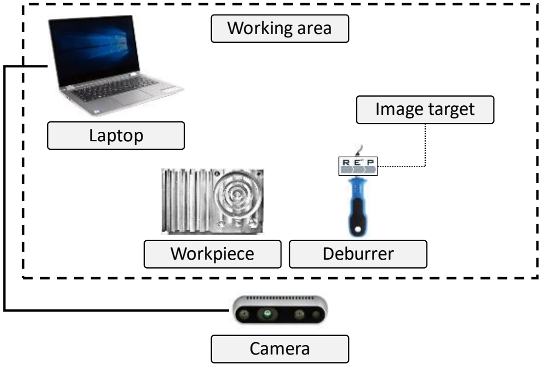


**Figure 13.** Schematic layout of the workplace

A deburring tool with a long shaft was used, which serves as an aid for object detection. An image target is attached to the shaft that references the cutting edge of the deburring tool for the detection software. This is necessary because when using a handheld deburring tool, a large section of the tool is hidden and thus there is no other placement option for an image target. Successful referencing via a model target is also not possible because of this. The attached target has an area of 19.6cm² and can be detected at a distance of up to about 60cm based on the preliminary tests.

As a workpiece for processing, a test component was manufactured as presented in Figure 14. The workpiece has straight and round contours, which allow determining the requirements of the created system in terms of complexity and precision.

The visual presentation of the control logics is preventive and instruction-oriented. When the targets are successfully detected, the referenced logics highlight the contours on which the deburring process must be executed. If the tool is moved over a corresponding spot, it will be marked as processed. Figure 15 shows the visualisation for straight lines and curves.

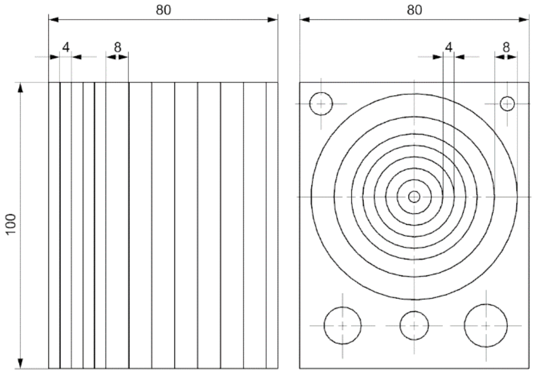


**Figure 14.** Test component, data in mm

The distance between the tool and the workpiece is used as control parameter. The logic trigger points are placed at intervals of approx. 10mm. They are triggered when the reference point of the tool falls below the threshold value of 3mm which is set in each control point. The value was determined by considering the results of the preliminary tests.

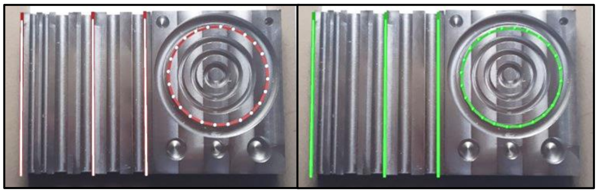


**Figure 15.** Information presentation: not triggered control logics (l.), fully triggered control logics (r.)

### 4.2 Implementation variants for workpiece detection

For the use case, three different variants for the target detection of the workpiece were implemented. The intention of these is to show and compare different possibilities of handling object detection for applications. Figure 16 shows the different principles.

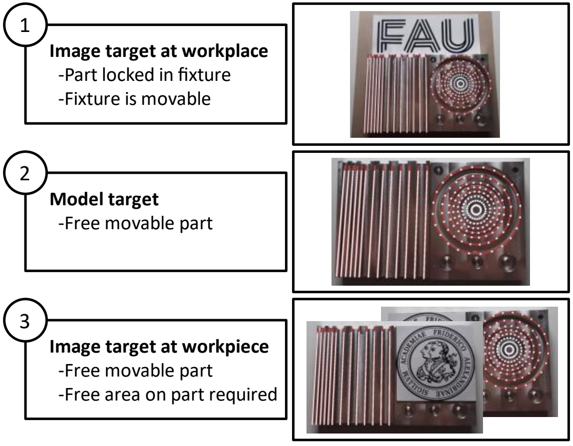


**Figure 16**. Implementation variants

In the first implementation variant, an image target attached to the workplace references the control logics. The workpiece is placed in a fixture that precisely defines its position relative to the image target, whereby the fixture is freely movable.

The second implementation variant detects the workpiece directly as a model target. The control logics are thus referenced via the model target and the component remains freely movable within the boundaries of a successful detection.

A third variant also uses an image target that references the control logic. In this case, the target is fixed on the workpiece, which allows it to be moved freely as long as the image target is detectable. However, a free area on the workpiece is required for attaching the target.

### 4.3 Experimental procedure

The functional features of the created system were evaluated by using the test component. The focus of the evaluation was to analyse the dynamic behaviour of the control logics respectively the object detection. For this purpose, the percentage of triggered control logics was measured for each of the created variants during an execution. A high percentage of triggered logics indicates a satisfactory functionality of the system, while a low trigger rate implies that the system is error-prone. Figure 17 shows a corresponding execution of the deburring process.

In addition to the variants created, the recording position (distance between camera and workpiece) as well as the speed of execution were affecting parameters for the system evaluation. The recording position was divided into three spherical ranges: a near range (up to 25cm), a middle range (25 to 50cm) and a far range (50cm to 75cm). The division of the execution speeds was made relative to a normal execution (factor 1), where the tool is moved with approx. 30mm per second during a working process. Four levels were defined: very slow (factor 0.25), slow (factor 0.5), normal (factor 1) and fast (factor 2) execution. For each combination of the affecting parameters, three measurements were taken.

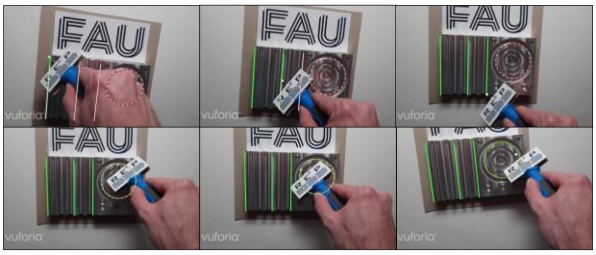


**Figure 17.** Example of an experimental procedure

### 4.4 Results and discussion

The results are shown in Figure 18. It illustrates a comparison between the different implementations in terms of execution speeds and distance ranges.

In general, it can be observed that the variants with image targets perform better than the version with a model target. In most of the cases, image target variants achieve values of more than 90% of triggered control logics, whereas with the model target version in the best case just 60% of the logics are triggered. One of the main reasons for this is that the workpiece used has highly reflective properties which makes it increasingly challenging for the software to detect the corresponding 3D contours of a model target. Even the simple display of the control logics on the workpiece shows noticeable displacements at certain points. With the customisable image targets, this difficulty is avoided at the cost of a necessary free area on the workpiece. In addition, it is generally simpler to detect characteristic 2D areas than 3D contours with 2D image recordings.

Regarding the detection range, it can be seen that the recording distance is a factor to be considered for this process, but its effects are comparatively small. With increasing distance, fewer logics are triggered, but the image targets used are sufficiently large so that the workpiece is adequately referenced over the entire distance range. With the model-target variant, a stronger influence can be seen due to the more difficult detection.

More significant is the effect of the execution speed. For all variants, a faster movement of the tool, and thus an increased dynamic within a scene, results in a detection deficiency. With increasing execution speed, control logic points are skipped more frequently and are, therefore, not triggered by mistake. This is due to the limited image capturing rate at which the system works for reasons of efficiency.

Another factor that must be considered, but which can only be slightly recognised from the measured values, is the hiding of targets or their features by, for example, tool handling. The system used to compensates this by saving and using the last detection coordinates in case of a detection loss of the target.

So, the implemented system is suitable for providing information about working steps that need to be executed for the deburring process in form of a position-related tutorial guidance, as well as a progress control of the process within the speed and distance limitations. For highly reflective surfaces, the system has reduced functionality when using model targets. However, an increase in functionality is expected for less reflective surfaces. The system is unsuitable for highly dynamic applications with fast motions.

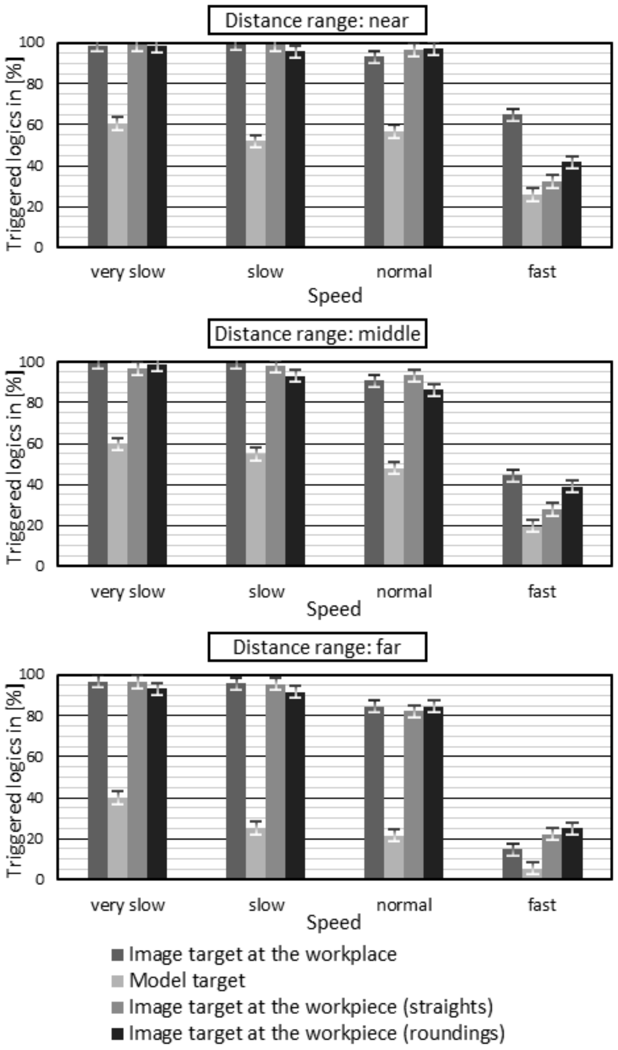


**Figure 18.** Comparison between the different variants

## 5 CONCLUSION AND OUTLOOK

Within this paper, a camera-based method is presented that supports manual contour and surface machining in the context of an AR application. The central idea is to monitor the process based on the object's movements. A target/actual comparison enables the evaluation of the execution and the provision of corresponding feedback. Therefore, the necessary position detection of the relevant objects is carried out using explicit feature matching methods. This is achieved with the help of image markers, which can be placed on the object or workstation, and model targets, which are represented by the object geometry itself. For the further processing of the received position data and the monitoring of complex manual operations, different control logics were introduced. The technical implementation is carried out with the help of the Unity 3D software and the AR software development kit from Vuforia. Vuforia provides functions for detecting the position of objects using image and model targets as well as extensive visualization options for an AR application. The developed control logics were implemented by application-specific programming and integrated as scripts in Unity 3D.

Several preliminary tests were conducted to evaluate the limitations and the precision of the developed approach. The focus was on the detection of objects using image markers and the triggering precision of position controls. The results were used to estimate the necessary target size for application in terms of the required detection distance. In addition, it was observed that with sufficient target detection control logics for position control work with deviations in the low millimeter range.

The validation of the developed approach was carried out based on the use case of manual deburring of a metallic component. A system was created that contains different object detection variants, with which developed position control logics for straight and round contours are referenced to a component to be processed. This makes it possible to provide process information on processing steps to be performed in the form of position-related tutorial instructions. In addition, the position control logics implemented can be used to perform a progress control of the process. In the case of strong reflections, functional limitations are to be expected for variants that use model targets. Further, the system is not designed for highly dynamic applications with fast movements.

Other inherent limitations of the system are, on the one hand, that a line of sight to the object must be given for its detection. If the line of sight is interrupted by occlusions or movement of the object out of the camera's field of view, the movement of the objects can no longer be tracked. On the other hand, only an indirect evaluation of the process execution is performed. Whether the desired state of a contour or surface has been achieved is not verified.

In summary, the created system can be classified as helpful for the manual execution of contour and surface processing. Unprocessed areas can be visualized to the operator, which helps to ensure complete processing.

However, there is still a need for further optimization. The central element of the system created is the detection and localization of objects. Therefore, the goal is to optimize the detection quality. A possible approach would be, for example, to carry out object detection redundantly by several systems. In addition, the research should focus on a further development of control logics. On the one hand, these can be extended regarding angle and speed monitoring. On the other hand, the inspection of more complex contours, which go beyond straight lines and circular shapes, can be implemented. This would improve the informative value of the indirect monitoring as well as expand the possible range of applications.

## REFERENCES

**[Beier 2015]** Beier, H.-M. Practical book on deburring technology. Guide to burr minimisation and burr removal for design and manufacturing. (in German, Praxisbuch Entgrattechnik. Wegweiser zur Gratminimierung und Gratbeseitigung für Konstruktion und Fertigung). 2015. 2., revised and extended version. Munich: Hanser.

**[Cui 2022]** Cui, X., Yu, M., Wu, L. Wu, S. A 6D Pose Estimation for Robotic Bin-Picking Using Point-Pair Features with Curvature (Cur-PPF). In: Sensors (Basel) 22 (5). 2022.

**[Digiacomo 2020]** Digiacomo, F. Afroz, A. S., Pelliccia, R., Inglese, F. Milazzo, M. Stefanini, C. Head-Mounted Standalone Real-Time Tracking System for Moving Light-Emitting Targets Fusing Vision and Inertial Sensors. In: IEEE Trans. Instrum. Meas. 69 (11), S. 8953–8961. 2020. DOI: 10.1109/TIM.2020.2995233.

**[Faccio 2019]** Faccio, M., Ferrari, E., Galizia, F. G., Gamberi, M., Pilati, F. Real-time assistance to manual assembly through depth camera and visual feedback. In: Procedia CIRP 81, S. 1254–1259. 2019. DOI: 10.1016/j.procir.2019.03.303.

**[Ferrari 2015]** Ferrari, R., Romano, F., Visentini, I. Q – Robot ROLL - Vision-based automatic deburring robot for cold billets. In: AISTech 2015 Iron and Steel Technology Conference and 7th International Conference on the Science and Technology of Ironmaking, S. 3072-3080.

**[Fukuda 2020]** Fukuda, K, Yamanobe, N., Ramirez-Alpizar, I. G., Harada, K. Assembly Motion Recognition Framework Using Only Images. In: 2020 IEEE/SICE International Symposium on System Integration (SII). 2020 IEEE/SICE International Symposium on System Integration (SII). Honolulu, HI, USA, 1/12/2020 - 1/15/2020: IEEE, S. 1242–1247.

**[Henke 2015]** Henke, J. : A methodology to increase value creation in the manual assembly of complex systems. (in German, Eine Methodik zur Steigerung der Wertschöpfung in der manuellen Montage komplexer Systeme). 2015.

**[Intel 2023]** intel (Hg.). Intel RealSense D435i. 2023. Available from https://www.intelrealsense.com/depth-camera-d435i/, Date of last actualization 31.05.2023.

**[Ittermann 2015]** Ittermann, P., Niehaus, J., Hirsch-Kreinsen, H. Working in Industry 4.0. Trend determinations and fields of action in labour policy (in German, Arbeiten in der Industrie 4.0. Trendbestimmungen und arbeitspolitische Handlungsfelder). 2015. Hans Böckler Stiftung: Düsseldorf.

**[Kellner 2020]** Kellner, F., Lienland, B. Lukesch, M. Production management (in German, Produktionswirtschaft). 2020. 2. updated and extended version. Berlin, Heidelberg: Springer Gabler.

**[Kinkel 2007]** Kinkel, S., Friedewald, M., Hüsing, B., Lay, G., Lindner, R. Working in the future. Structures and trends in industrial work (in German, Arbeiten in der Zukunft. Strukturen und Trends der Industriearbeit). Office of Technology Assessment at the German Parliament (in German, Büro für Technikfolgen-Abschätzung beim Deutschen Bundestag). 2007.

**[Manuri 2019]** Manuri, F., Pizzigalli, A., Sanna, A. A State Validation System for Augmented Reality Based Maintenance Procedures. In: Applied Sciences 9 (10), 2019. S. 2115. DOI: 10.3390/app9102115.

**[Mühlbauer 2022]** Mühlbauer, M., Kutzner, K., Sommer, A., Würschinger, H., Hanenkamp, N. An Approach to Progress Monitoring of Industrial Manual Processes Based on Camera Recordings and Object Interactions. In: Procedia CIRP 107, S. 582–587. 2022. DOI: 10.1016/j.procir.2022.05.029.

**[Princely 2014]** Princely, F. L., Selvaraj, T. Vision Assisted Robotic Deburring of Edge Burrs in Cast Parts. In: Procedia Engineering 97, S. 1906–1914. 2014. DOI: 10.1016/j.proeng.2014.12.344.

**[Roth 2020]** Roth, E., Moncks, M., Bohne, T., Pumplun, L. Context-Aware Cyber-Physical Assistance Systems in Industrial Systems: A Human Activity Recognition Approach. In: 2020 IEEE International Conference on Human-Machine Systems (ICHMS). 2020 IEEE International Conference on Human-Machine Systems (ICHMS). 2020. Rome, Italy, 07.09.2020 - 09.09.2020: IEEE, S. 1–6.

**[Rudorfer 2021]** Rudorfer, M. Towards robust object detection and pose estimation as a service for manufacturing industries. Technische Universität Berlin. 2021.

**[Sahin 2020]** Sahin, C., Garcia-Hernando, G., Sock, J., Kim, T.-K. A review on object pose recovery: From 3D bounding box detectors to full 6D pose estimators. In: Image and Vision Computing 96, S. 103898. 2020. DOI: 10.1016/j.imavis.2020.103898.

**[Schäfer 1975]** Schäfer, F.: Deburring: Theory, procedures, equipment (in German, Entgraten: Theorie, Verfahren, Anlagen). Mainz: Krausskopf. 1975.

**[Schenk 2015]** Schenk, M. Production and logistics in future (in German, Produktion und Logistik mit Zukunft. Berlin, Heidelberg: Springer Berlin Heidelberg. 2015.

**[Song 2013]** Song, H.-C.; Song, J.-B.201 Precision robotic deburring based on force control for arbitrarily shaped workpiece using CAD model matching. In: Int. J. Precis. Eng. Manuf. 14 (1), S. 85–91. 2013. DOI: 10.1007/s12541-013-0013-2.

**[Stan 2022]** Stan, L., Nicolescu, A. F., Pupăză, C., Jiga, G. Digital Twin and web services for robotic deburring in intelligent manufacturing. In: Journal of intelligent manufacturing, S. 1–17. 2022. DOI: 10.1007/s10845-022-01928-x.

**[Tellaeche 2016]** Tellaeche, A., Arana, R. Robust 3D Object Model Reconstruction and Matching for Complex Automated Deburring Operations. In: J. Imaging 2 (1), S. 8. 2016. DOI: 10.3390/jimaging2010008.

**[Yan 2020]** Yan, W., Xu, Z., Zhou, X., Su, Q., Li, S., Wu, H. Fast Object Pose Estimation Using Adaptive Threshold for Bin-Picking. In: IEEE Access 8, S. 63055–63064. 2020. DOI: 10.1109/ACCESS.2020.2983173.

**[Yang 2020]** Yang, C.-L., Li, W.-T., Hsu, S.-C. Skeleton-based Hand Gesture Recognition for Assembly Line Operation. In: 2020 International Conference on Advanced Robotics and Intelligent Systems (ARIS). Taipei, Taiwan, 8/19/2020 - 8/21/2020: IEEE, S. 1–6. 2020.

**[Zhang 2019]** Zhang, Z., Zhou, Y., Liu, H., Gao, H. In-situ water level measurement using NIR-imaging video camera. In: Flow Measurement and Instrumentation 67, S. 95–106. 2019. DOI: 10.1016/j.flowmeasinst.2019.04.004.

**[zur Jacobsmuhlen 2013]** zur Jacobsmuhlen, J., Kleszczynski, S., Schneider, D., Witt, G. High resolution imaging for inspection of laser beam melting systems. In: 2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC): IEEE. 2013

## CONTACTS:

Matthias Mühlbauer, M. Sc.
Friedrich-Alexander-University Erlangen-Nürnberg, Chair of Resource and Energy Efficient Production Systems
Dr.-Mack-Str. 81, Technikum 1, D-90762 Fürth, Germany
e-mail: matthias.muehlbauer@fau.de
website: https://www.rep.tf.fau.de/