Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 104 (2021) 1113–1118

www.elsevier.com/locate/procedia

54th CIRP Conference on Manufacturing Systems

Image-based state tracking in Augmented Reality supported assembly operations

Vasilios Zogopoulos[a,*], Merwan Birem[a], Roeland De Geest[a], Robbert Hofman[a], Lode Jorissen[b], Bram Vanherle[b], Dorothy Gors[a]

[a]*Flanders Make, Gaston Geenslaan 8- B-3001 Leuven, Belgium*
[b]*Hasselt University - tUL -Flanders Make, Expertise Centre for Digital Media- Wetenschapspark 2 - 3590 Diepenbeek, Belgium*

* Corresponding author. Tel.:+32-016910690. *E-mail address:* vassilis.zogopoulos@flandersmake.be

Abstract

Visual tracking and holographic information representation techniques have become robust enough to support operators in complex tasks on the shop floor. This paper presents an approach for coupling AR-supported assembly task instructions with image-based state tracking, so as to assist the operators in product assembly operations. The developed system consists of a visualization platform for AR-supported assembly instructions, a state tracker that includes object recognition, localization and hand tracking, using deep neural networks, and a server that handles the data exchange between the two. The developed framework is applied and validated in an industrial use case.

Keywords: Augmented reality; Assembly; State tracking; Neural network; Object recognition

1. Introduction

As manufacturing is characterized by large production rates as well as high number of product variants and personalization features, product quality monitoring becomes increasingly challenging in modern production [1]. Especially, as flexible manufacturing requires human operators in the loop, the development of digital systems that track the state of the task are required.

Non-destructive testing has been the most used method to track product quality in the production, without creating disruptions, especially useful in the case of customized production where the product batches are divergent. More specifically, computer vision techniques have arisen in the industry and are nowadays commonly used for detecting, tracking products in the product line as well as monitoring their quality.

Computer vision and more specifically environment tracking has been the one of the core components for another prominent technology which has proven to be useful in manufacturing; Augmented Reality (AR). AR allows the projection of task-related information in the field of view of the operator. As tracking algorithms has increased the precision in the tracked environment, and new methods that maximize the utilization of digital content generated in parts of manufacturing process, such as product design, an increasing number of AR applications are reaching the shopfloor.

Summarizing, this study presents an approach for combining image-based stare tracking in AR supported assembly tasks. The state tracker combines hand, part and tool detection and tracking so as to maximize the accuracy of tracking task progress. It is capable of providing real-time updates to the AR instructions application, together with dynamic information from the environment of the operator, such as part and tool position. To validate the applicability of the developed system in an industrial scenario, it is applied in a compressor assembly, while the operator is supported by the developed state tracking system.

10.1016/j.procir.2021.11.187

2. State of the art

As product variants are rapidly increasing, and the manufacturing era is characterized by mass customization, it is important to deploy mechanisms in the shop floor that will contribute to increased flexibility [2]. Human operators are a factor that offers this desired level of flexibility (compared to automated systems). Though, their role has drastically shifted from doing repetitive task towards continuously altering instructions and collaboration with robots [3]. As reconfigurability has become a significant aspect of modern manufacturing, it is important to provide operators with information on the right time and in a perceivable way so that they can make the right decisions and be more efficient [4]. Many recent studies have explored new human-centered approaches that integrate Industry 4.0 technologies so as to facilitate the operators [5].

One of the potential shapes of the operator in the Industry 4.0 era is the Augmented operator, where the operator's capabilities are empowered by an Augmented Reality (AR) system [6]. AR provides assistive solutions in different industrial fields making information quickly available in a perceivable and immersive way. The introduction of this technology in manufacturing has shown merit in reducing the human error, enabling a new way of educating people and increasing the collaboration [7]. For this purpose, plenty of AR applications have been developed to support the operators in assembly tasks [8], maintenance [9] as well as in processes that require machine operation [10]. As this technology is empowered from information already available in other digital systems in the manufacturing system, integrated frameworks that connect multiple components have emerged in the literature, highlighting its potential in modern manufacturing [11].

Another technology based on visual tracking that is trending in manufacturing is operator tracking, either the full body or specific parts (e.g. hand tracking). Image-based skin detecting is a critical processing step for many applications, including hand detection and gesture recognition. In our context of state detection, gives the hand-tip and -joint location valuable information about the actions of the operator. Chyad, Alsattar and al. [12] give an overview of the various techniques to recognize skin. They categorize different techniques based on the used color space, technique and method. The skin tone can be identified in different color spaces, where HSV and YCBCr are the most popular ones. Common techniques for skin detection, or in extension hand-joint or gesture recognition (Gurav and Kadbe [13]), are neural networks, Gaussian mixture models, Support Vector Machines and Adaptive Boosting. As methods, Chuad and Alsattar distinguish four categories: pixel-based, region-based, hybrid and other methods [12].

Based on the increased need for systems that support the operators in the production line, this paper proposes a system that combines AR assembly instructions with image-based state tracking. State tracking is enabled via object and hand tracking on different levels of the assembly task. Moreover, data exchange between the state tracker and the AR visualization application dynamically provide the operators with information about the parts in their environment and also automatically detects correct assembly step completion.

3. System description

The overview of the proposed system is depicted in Fig. 1 below. Image feed from the available sources is fed to the state tracker where the current status of the process is evaluated with a certainty value. The estimated score is then pushed to the flow manager, where task completion confidence is calculated. The flow manager also knows which parts of the assembly are associated to which task, and keeps track of the tasks that are already finished or still need to be executed. Each sub-component and its role in understanding the current situation is described in detail in the following sections.

3.1. State tracker

The state tracker consists of two parts: the first is the low level block that process the data coming from sensors (e.g. top view cameras 2D or 3D, body or head worn cameras) and extract the needed information from the acquired data, information like object location, hands and parts position. The

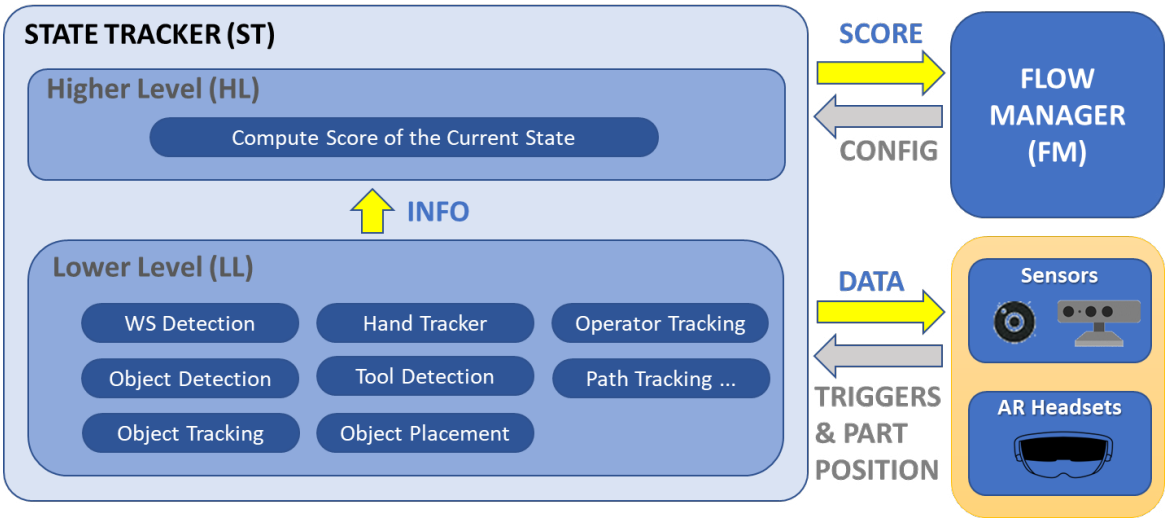

Fig. 1. An overview of the state tracker sub-blocks and the data exchanged between the sensors and the flow manager layer.

second part of the state tracker is the higher level block that uses the information extracted in the low level to evaluate the state of the assembly and to compute a confidence score (between 0 and 1) that reflect the completion of the ongoing step. This score is a combination between the estimation that the part and/or tool is picked (Section 3.2 and 3.3) and that it is positioned in the final position (analyzed in Section 3.4). The state tracker feeds those two scores to the "Flow Manager" continuously, where if the score surpasses the pre-set threshold, the step is marked completed.

In Fig. 1, an overview of the state tracker is presented. The lower level part of the state tracker is composed of several sub blocks used to detect/track the hand of the operator, detect/track the object that will be manipulated, confirm picking and placement operations, tool detections, path tracking and surface treatment (described in the following Sections). Note that depending on the use case and the assembly steps requirements some of the sub modules will not be active. Each image tracking system (described in Sections 3.2-3.5) gives a confidence (from 0 to 1) that each movement is performed correctly. Confidence may sometimes not be 1, despite the fact that the step is completed, because of occlusions (e.g. the hand covers the camera's view to the part and the part detection gives a lower score). The higher level part of the state tracker uses the information extracted from the received data, which is the output of the sub-blocks of the lower level together with the historical data. Ultimately, the state tracker uses the information received so as to keep track of the state of the assembly action, the position of the parts in the workspace and to estimate if the task has been completed.

3.2. Object detection & tracking

At first, the workstation layout corresponding to the camera used needs to initialize. In this sub-block, a marker-based algorithm (e.g. arUco marker [14]) is implemented for the initialization phase to load the layout of the work station, the assembly steps (i.e. work instruction), the components, tools and their location relatively to the marker. During the assembly process an important information to provide to the operator with, is which object to pick and where to pick it. This sub-block of the state tracker is used with two different modes to indicate what and where is the object of interest during each assembly step.

The first mode to recognize the object of interest is based on a deep learning neural network trained on a synthetic database generated using the CAD file(s) of the object(s). The process of training is done upfront for each geometry in each step (i.e. offline which allows to generate a large dataset and finetune the object detector further). The training of the object detector is done by "Transfer Learning" using the YOLOv3 architecture [15]. The second mode is based on picking from a storage bin from a fixed and known location. This mode is used when no CAD file of the object is available, or if it is difficult to design an accurate object detector for it, due to size, shape, texture or reflectivity (e.g. washers, bolts and screws).

Next to the object detector, an object tracker module is used to track the location of the object of interest over time. The tracking of the object is done based on its visual appearance, movements and the hand of the operator.

3.3. Hand tracking

Additionally, the operator's hands are being continuously monitored. Even if object picking, tracking and placement recognition fails, the operator action can be supervised by the movement of his hands. For this, we devised an adaptive intensity modeling algorithm to track the hands. Based on the hands' previous movements, a rough estimation of the current hand position is made. Whereafter, the hand contour is delineated within that estimated region using an intensity model for the skin tone. This model is updated each frame using the distribution of a RGB-derived intensity measurement within the detected hand contour. By tracking the hands we can monitor if the operator went to the step-specified pick and place region, which gives an indication whether a object picking and placing could have occurred.

3.4. Object Placement Detection

To detect whether a part of the assembly has been put in place, the camera image is compared with a render from the CAD model, generated at the same position as the camera. The rendered image contains all the steps that have been completed, as well as the parts that need to be placed in the current step. Lists of completed steps and steps that are currently being processed are provided by the flow manager. The pose of the camera, used to render the image from the same viewpoint, can be obtained in several ways: by using a predefined marker, by Simultaneously Localization and Mapping (SLAM) or by values reported by the camera device's sensors (e.g. in case of a head mounted display). The object has been placed if the camera image matches with the rendered image. Since we don't want to compare the full scene, we only compare the regions in the images that should contain the objects. This region can be obtained by taking the bounding box of the projection of the parts that need to be placed.

Due to limitations of the CAD model and due to differences between the real world and the virtual scene, one cannot directly compare the images. Instead, we first apply a canny edge detector on both images. This detector creates an edge-map in which most of the shading effects as well as any color differences in the images are removed. Next, a Histogram of Oriented Gradients (HOG) Descriptor is calculated for both edge-maps. A HOG descriptor divides the image in cells and calculates a histogram of the image gradient observed in each cell. As such it contains a rough description of the object's structure. Finally, the Normalized Cross-Correlation (NCC) of both descriptors is calculated. We assume that the parts are in place if the NCC is higher than a given threshold value. This process is shown in Fig. 2.

To increase stability of this method, the camera image is compared not only to a CAD render that contains all previous steps and the current step, but also to the image to a CAD render that only contains all previously completed steps.

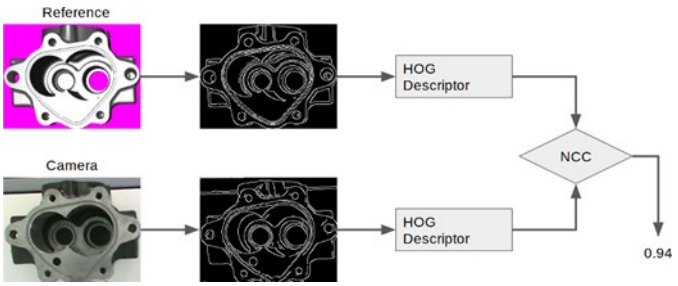

Fig. 2. Detecting whether an object is in place. A reference image is rendered from the same position as where the real camera is positioned. Edge maps are calculated for both images and a HOG descriptor is calculated for both approaches. The resulting descriptors are compared to each other using Normalized Cross-Correlation.

By comparing the corresponding NCC's, an additional measure is taken to see whether the parts have been placed: the NCC of the former comparison should be larger than the one of the latter, since this means that the current state of assembly is more similar to the render with the step completed than to the one without. This is useful in cases where the scene containing the part is similar to the scene without it. An example of this is a bearing that needs to be placed in a matching hole.

Stability can be further increased by checking whether the correct tools (e.g. screwdriver or wrench, but also the fingertips of the operator), have been used on the area where the objects need to be placed. This can be done using the tool landmark detection (described in Section 3.5). To detect the landmarks of the user's hands we utilize the joint detection network from Zhou et al. [16]. In the case that these tools are the hands of the operator, we take the average of the landmarks at the tips of the thumb and index fingers to check interaction with the parts, since objects are usually picked up using those two fingers.

The explained approach has the advantage that it requires no extensive training to detect the required objects. It works well on distinctive objects, and can be made to work for less distinctive objects, as explained above. However, it requires that the poses of the assembly station and the camera are known at all times.

3.5. Path Tracking and Surface Treatment

Some steps require that a tool follows a predefined path or that it is used to treat a surface. Examples include applying glue or primer on a surface or cleaning the surface. To check if such a procedure has been completed, we define 3D points on the object. These points make up the path or surface that needs to be treated. Every point is given a radius in which the tool is allowed to operate. Since the landmarks that we calculate for the tools only give us 2D locations in the camera image, we project the 3D points and their radius onto the image plane. Interaction between a path point and a landmark point of a tool can be checked by calculating the distance between them and comparing it to the projected radius.

Depending on the kind of treatment, the order in which one is allowed to handle the points may be different. For a surface treatment for which no specific order is required, for example cleaning a surface, the system marks every point with which the tool has interacted as "treated". When a specific order is required or expected, for example when applying glue on a specific path, the 3D points should be defined in the correct order. For every frame interaction, only the first "untreated" point in the list should be checked. This point should be marked as "treated" if there is interaction. This procedure returns a score indicating the percentage of points that have been treated.

Additional information, such as a timestamp or an indication of velocity, may be provided with the points to give additional details to the operator.

3.6. Tool detection

There are certain actions in the assembly sequence that require the use of a tool to be completed: placement steps where for which a tool is used (e.g. tightening a bolt after placing it) or surface treatment steps (e.g. apply glue to a surface using a glue gun). To verify the completement of these steps we need to know the location of certain keypoints of the tool in question in the input image. These 2D locations can then be used to approximate whether there has been contact with the target part or surface region by checking for overlap on the image.

Since it is known what tool will be used for each step and approximately in which part of the image, no object detection is used to identify tools. Instead, the region around the target part or surface is directly used as input for a keypoint detection algorithm. Keypoint detection is achieved by a neural network that, given an image, outputs a set of heatmaps. A heatmap is generated for each type of keypoint. Such a heatmap indicates, for each pixel, the probability that the keypoint is at location.

The neural network used is a Fully Convolutional U-Net [17] that uses MobileNetV2 [18] as a feature detector. Training images are generated using Unity's High Definition Rendering Pipeline. From the ground truth keypoint locations of these images, heatmaps are created by evaluating a Gaussian around the location. The feature detection network is initialized on weights trained on ImageNet, the weights are then frozen to be able to detect features from real life images [19]. Keypoint locations are retrieved from the network output by finding local maxima above a certain threshold, as shown in Fig. 3. A new network is trained for each tool that is used in the process.

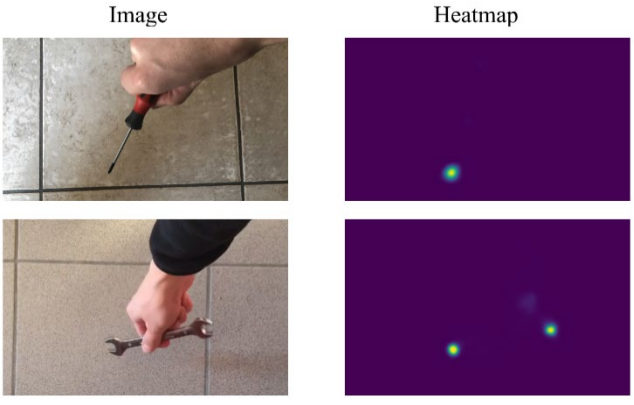

Fig. 3. Detecting the 2D locations of the keypoints on an image of a tool. On the left side the raw input image is shown. The heatmap generated by the neural network is shown on the right.

3.7. AR assembly instructions

The state tracker is connected with an application that provides assembly instructions to the operator. The instructions are generated based on the product's CAD file and broken down to assembly steps using an algorithm that evaluates the capability of each of the remaining parts to move along one of its six principal directions [20]. In each step of the assembly, the application loads the corresponding part and if the part is a fastener, a tool (e.g. a wrench in the case of a bolt) and a rotational arrow are also visualized, giving the operator a more immersive feeling. The application, apart from the instructions on what to assemble where, is also exchanging information with the state tracker. When the assembly moves to a new step, the position of the part to be picked is highlighted. If the part is on the worktable a green plane is visualized below it; if it is in a toolbox the box is highlighted in green color. Moreover, when the state tracker detects that the part is in its final position, it sends a trigger to the application to move to the next step.

4. System implementation

At the core of the state tracking system is a ROS network of nodes that communicate by publishing on and listening to ROS topics. Thanks to the use of the ROS framework, the system is modular: when a new node is introduced, it can simply subscribe to the topics it needs to obtain information from, and publish its outputs on topics that other nodes have subscribed to. The broader goal of the system is to estimate the context in which the operator is located. More specifically, the goal is to estimate the state of the assembly the operator is performing. To that end, input data is processed from one or several imaging sources; in the simplest case, an overhead camera. The images produced by this source are published on a ROS topic that any other ROS node can subscribe to.

Subscribers of the image topic include an object detection node and an object and hand tracking node. The former detects where the relevant parts are, the latter tracks them as they move across the 2D image. Their 2D image space coordinates are published. Image processing nodes are mostly unaware of the state of the assembly. They merely process images in order to provide information to a node that keeps track of the state, conveniently called the "state tracker" node. This node is configured to keep track of the steps pre-defined in an automatically generated instruction document. It makes an estimate for the likelihood that a step has been completed, based on the information of the image processing nodes. For example, when the no hand has been detected to be in the neighborhood of the part relevant for this step, the likelihood will remain low.

Outside of this core network of ROS nodes are some other software blocks or systems that interact with the ROS network through simple web requests or through a bridge: both a TCP bridge and a WebSocket bridge have been implemented to enable communication with outside systems. Examples of such outside systems include a graphical user interface implemented in the Qt framework, a web based graphical user interface implemented in React, and an AR application for HoloLens (2).

Another one of those outside systems is a database called "content repository", which enables the centralized storage of instruction documents, visualization assets and configuration parameters. By centralizing all data, the software systems remain highly configurable and are guaranteed to remain synchronized. The communication to this content repository is implemented by standard HTTP requests to a REST API, and the data is structured according to the ISA-95 standard. As a result, other systems that adhere to those standards are able to communicate to the same content repository.

5. Validation

The developed system is applied in an industrial use case that revolves around the assembly of a compressor. As human operators are an important part of the assembly sequence, introducing new ways to integrate digital instructions will facilitate and accelerate the instructions generation stream, while also the automated state tracking will reduce the errors of the operators and boost their confidence in cases with limited experience. Especially since the operators are called to deal with different product variants that require high attention to detail, the AR instruction, together with the state tracker aim to provide accurate and constantly updated instructions.

Based on the CAD files, the assembly sequence is generated. In each step, the part/ sub- assembly that needs to be assembled is separated, together with position related (final position, assembly direction) and tracking related data (way of tracking, tool). Based on the assembly sequence, the AR instructions are generated.

To track the scene, the feed of a top-down camera is used. In each step, the part to be picked is detected and its position is sent to the AR application. The part location is highlighted, either by projecting a green square on the working table (Fig.4d) or by highlighting in green its position in the toolbox if its position is preregistered (Fig. 4b). Some examples of how the assembly instructions are visualized in AR, considering also the usage of the required tools, are presented in the Fig. 4 below.

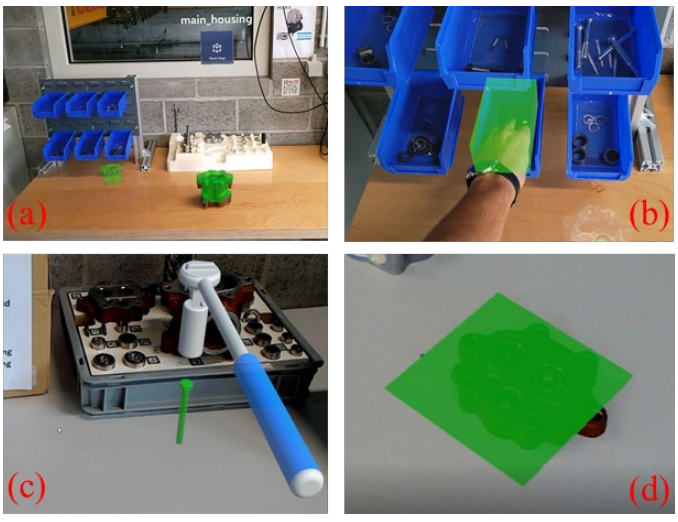

Fig. 4. The key features of the AR assembly support application: a) instructions visualization, b) hand tracking for parts in the toolbox, c) tool visualization, d) part to be picked is highlighted

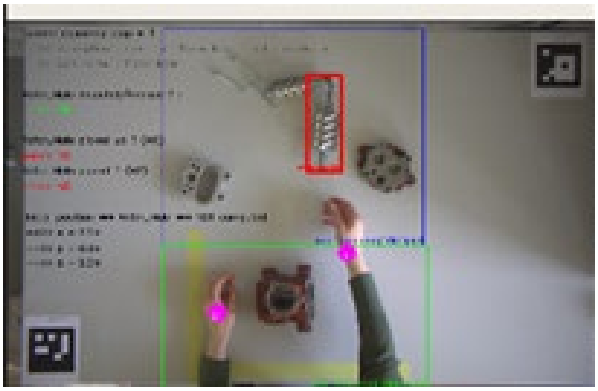

Fig. 5. The state tracker monitors the process. Red area: Part to be picked detection. Pink dots: operator's hands detection

As the operator moves towards the part to be picked and the picking confidence surpasses the set threshold, the state tracker then sends the trigger to the AR application to proceed to the assembly instructions. The part is visualized moving towards its final position with an automatically created animation, based on the data coming from the CAD analysis. The top-down camera tracks the motion of the hand together with the part until it reaches its final position, as it is depicted in the Fig. 5. When the step was detected as completed (threshold exceeded), the flow manager sends a trigger to the state tracker and the AR application to proceed to the next step.

6. Conclusions and future work

This study presents a state tracker based on object, hand and tool recognition that supports automatic detection of task completion in manual assembly operations. The developed state tracker is used together with an AR application that aims to support the operator by providing high fidelity instructions in a semi-automated way. The state tracker allows the operators to focus on the task, while also increasing their confidence that the task has been successfully completed.

As a future development, the training of the object recognition algorithms can be solely supported by a sophisticated photorealistic image generator, so that the training dataset is generated in a virtual environment. Additionally, the developed framework could record limiting cases (whenever the operator confirms that a step is completed though the algorithm was not that confident) so as to improve its training.

Acknowledgements

This research is supported by VLAIO (Flanders Innovation & Entrepreneurship) Flanders Make, the strategic center for the manufacturing industry in Flanders, within the framework of the FAMAR ICON-project (HBC.2018.0249).

References

[1] Mourtzis D, Doukas M. The evolution of manufacturing systems: From craftsmanship to the era of customisation. InHandbook of research on design and management of lean production systems 2014. p. 1-29. IGI Global

[2] Wang Y, Ma HS, Yang JH, Wang KS.. Industry 4.0: a way from mass customization to mass personalization production. Advances in Manufacturing 2017:5(4): 311-320.

[3] Dimeas F, Fotiadis F, Papageorgiou D, Sidiropoulos A, Doulgeri Z. Towards progressive automation of repetitive tasks through physical human-robot interaction. In Human Friendly Robotics 2019. P. 151-163. Springer, Cham.

[4] Nelles J, Kuz S, Mertens A, Schlick CM. Human-centered design of assistance systems for production planning and control: the role of the human in industry 4.0. 2016 IEEE Int Conf Ind Technol 2016: 2099–2104. https://doi.org/10.1109/ICIT.2016.7475093

[5] Romero D, Bernus P, Noran O, Stahre J, Fast-Berglund Å. The operator 4.0: human cyber-physical systems & adaptive automation towards human-automation Symbiosis work systems. In: Nääs I, Vendrametto O, Mendes Reis J et al (eds) Advances in production management systems. Initiatives for a sustainable world. Springer International Publishing, Cham, 2016; p. 677–686

[6] Benešová A, Tupa J. Requirements for education and qualification of people in industry 4.0. Procedia Manuf 2017;11:2195–2202. https://doi.org/10.1016/j.promfg.2017.07.366

[7] Esengün M, İnce G. The Role of Augmented Reality in the Age of Industry 4.0. In Industry 4.0: Managing The Digital Transformation: Springer, Cham; 2018. p.201-215. doi: 10.1007/978-3-319-57870-5_12

[8] Mourtzis D, Zogopoulos V, Xanthi F. Augmented reality application to support the assembly of highly customized products and to adapt to production re-scheduling. The International Journal of Advanced Manufacturing Technology 2019;105(9):3899-3910.

[9] Mourtzis D, Vlachou A, Zogopoulos V. Cloud-based augmented reality remote maintenance through shop-floor monitoring: a product-service system approach. Journal of Manufacturing Science and Engineering, 2017;139(6).

[10] Mourtzis D, Zogopoulos V, Katagis I, Lagios, P. Augmented Reality based Visualization of CAM Instructions towards Industry 4.0 paradigm: a CNC Bending Machine case study. Procedia CIRP 2018;70: 368-373.

[11] Mourtzis D, Vlachou E, Zogopoulos V, Fotini X. Integrated production and maintenance scheduling through machine monitoring and augmented reality: An Industry 4.0 approach. In: IFIP Int. Conf. on Adv. in Prod. Mngmt Syst.: Springer, Cham;2017. p. 354-362. doi: 10.1007/978-3-319-66923-6_42.

[12] Chyad MA, Alsattar HA et al. The Landscape of Research on Skin Detectors: Coherent Taxonomy, Open Challenges, Motivations, Recommendations and Statistical Analysis. IEEE Access. 2019;7.

[13] Gurav RM, Kadbe P.K. Real time finger tracking and contour detection for gesture recognition using OpenCV. 2015 International Conference on Industrial Instrumentation and Control (ICIC) College of Engineering Pune. India. May 28-30, 2015; p. 974-977

[14] Garrido-Jurado S, Muñoz-Salinas R, Madrid-Cuevas FJ, Marín-Jiménez MJ. Automatic generation and detection of highly reliable fiducial markers under occlusion". Pattern Recogn. 2014;47(6): 2280-2292. DOI=10.1016/j.patcog.2014.01.005

[15] Redmon J, Farhadi A. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018).

[16] Zhou Y, Habermann M, Xu W, Habibie I, Theobalt C, Xu F. Monocular Real-time Hand Shape and Motion Capture using Multi-modal Data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020;p. 5346-5355.

[17] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation, in MICCAI 2015

[18] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted Residuals and Linear Bottlenecks, in Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[19] Hinterstoisser S, Lepetit V, Wohlhart P, KonoligeK. On Pre-Trained Image Features and Synthetic Images for Deep Learning," in European Conference on Computer Vision, 2018.

[20] Gors D, Put J, Vanherle B, Witters M, Luyten K, Semi-automatic extraction of digital work instructions from CAD models. Procedia CIRP; (In press)