



# Enhancing Process Data in Manual Assembly Workflows

Sönke Knoch<sup>(✉)</sup>, Nico Herbig, Shreeraman Ponpathirkoottam, Felix Kosmalla, Philipp Staudt, Peter Fettke, and Peter Loos

German Research Center for Artificial Intelligence (DFKI),  
Saarland Informatics Campus, Saarland University, Saarbrücken, Germany  
`soenke.knoch@dfki.de`

**Abstract.** The rise of Industry 4.0 and the convergence with BPM provide new potential for the automatic gathering of process-related sensor information. In manufacturing, information about human behavior in manual assembly tasks is rare when no interaction with machines is involved. We suggest technologies to automatically detect material picking and placement in the assembly workflow to gather accurate data about human behavior. For material picking, we use background subtraction; for placement detection image classification with neural networks is applied. The detected fine-grained worker activities are then correlated to a BPMN model of the assembly workflow, enabling the measurement of production time (time per state) and quality (frequency of error) on the shop floor as an entry point for conformance checking and process optimization. The approach has been evaluated in a quantitative case study recording the assembly process 30 times in a laboratory within 4 h. Under these conditions, the classification of assembly states with a neural network provides a test accuracy of 99.25% on 38 possible assembly states. Material picking based on background subtraction has been evaluated in an informal user study with 6 participants performing 16 picks, each providing an accuracy of 99.48%. The suggested method is promising to easily detect fine-grained steps in manufacturing augmenting and checking the assembly workflow.

**Keywords:** Manual assembly · Computer vision · BPM · Industry 4.0

## 1 Introduction

The current trend of automation and data exchange known under the term Industry 4.0 [6, 9] addresses the convergence of the real physical and the virtual digital world in manufacturing. It comprises the introduction of cyber-physical systems (CPS), Internet of Things (IoT) and cloud computing in a fourth industrial revolution, where manufacturing companies face volatile markets, cost reduction pressure, shorter product lifecycles, increasing product variability, mass customization leading to batch size 1 and, with rising amounts of data, developments towards a smart factory [1].

To plan, construct, run, monitor and improve a flexible assembly work station or CPS tackling the challenges of Industry 4.0, engineers and managers require detailed information about the assembly workflow in the life-cycle phases of a CPS, e.g. [16]. In workflows with manual tasks, this information contains data on human behavior, such as grasp distance, assembly time and the effect the workplace design has on the assembly workflow. It can be used to plan and construct efficient assembly work stations and receive information on the executed workflows to establish a continuous improvement process (CIP/Kaizen) within the organization. The data has to be approximated or gathered manually consuming time and money.

We see a large potential in the technical support and automatization of data gathering processes providing information about a workflow's execution regarding time and quality. The accurate detection of human behavior during the assembly workflow, and as a consequence, the generation of meaningful process logs about that workflow, is a challenging task. On the hardware side, the selection of appropriate sensors, their integration into the assembly system and the robustness against the conditions in the manufacturing context have to be considered. On the software side, these sensors have to be used to detect activities reliably, deliver results fast and provide complete information in a homogeneous data format. If the physical setup, the software configuration, and the operation of such a system support workflow execution efficiently, this will add value to the organizations deploying them: (1) Process models will be enriched with detailed information about the assembly steps. (2) Live workflow tracking enables performance checking online or offline. (3) Further analysis of workflow traces can be used to adapt and optimize the workflow execution.

In a first iteration, an artifact, fast and easy to set up in terms of configuration and instrumentation, was developed integrating multiple sensors to analyze hand and body posture, as well as material picking. For this we used ultrasonic sensors, infrared, RGB+Depth and RGB cameras [7]. This proof-of-concept implementation was then extended and evaluated in a case study with 12 participants in a second iteration [8]. In this work, we reduce the sensor setting to the most promising sensors (2 RGB cameras and 1 hand sensor) applying new methods from computer vision and machine learning to achieve a high resolution in recognizing assembly tasks. Further on, events about task-related events are correlated to process tasks in a BPMN model. Now, manual assembly workflows are controlled by a model that can be created, configured and adapted in a graphical BPMN editor to deploy new workflows, facilitate improvements and support the worker appropriately, thereby achieving a new level of flexibility.

## 2 Related Work

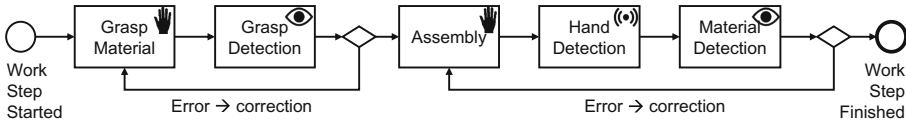
Related work can be found at the intersection of BPM with cognitive computing, context-awareness and human activity recognition, here mainly vision-based.

*Cognitive BPM* comprises the challenges and benefits of cognitive computing in relation to traditional BPM. Hull and Motahari Nezhad [4] suggest a

cognitive process management system (CPMS) to support cyber-physical processes enabled by artificial intelligence (AI). Marrella and Mecella [11] propose the concept of such a CPMS which automatically adapts processes at run-time, taking advantage of the AI's knowledge representation and reasoning. Similar to our system the alignment between real physical and expected modeled behavior can be measured. In *context-aware BPM*, the user behavior is set in relation to situations affecting the behavior. Transferred to our approach, user behavior corresponds to detected worker activities in the context of a concrete assembly task from a process model instance executed at a concrete assembly work station. Jaroucheh, Liu, and Smith [5] apply linear temporal logic and conformance checking from process mining to compare real with expected user behavior. Since the collection of high resolution data for the aforementioned systems is not a trivial task, augmenting the process with information from cognitive computing platforms is the focus within this research activity.

*Vision-based human activity recognition (HAR)* mainly focuses on 2D video data and applies various machine learning methods (for an overview see [12]). HAR applications in manufacturing are sparse but occur more frequently since the rise of IoT, industrial internet and Industry 4.0. Within the field of human robot collaboration, [10] use 3D video data to determine the hand position of a worker sitting in front of an assembly work table and collaborating with an assistive robotic system applying Hidden Markov Models (HMM). Two cameras are mounted to the assembly workstation and are calibrated to each other. High calibration effort facilitates the application of the method only in restricted set-ups. Similarly, [13] apply hierarchical HMMs in a multimodal sensor setting within the same domain. They combine RGB-D (Kinect 2) and IR (Leap Motion) cameras to detect fine-grained activities (e.g. assembly, picking an object, fixing with a tool) and gestures (e.g. pointing, thumb up) and analyze the results of the sensors respectively and in combination. Combination of sensors show the best average recognition results for activities in most cases. Unlike these approaches, focusing on gesture detection, we focus on the detection of assembly states in form of image classes allowing the connection to state transitions in process models. It is an example how AI can be applied to ensure process quality and adherence to time constraints.

In the field of *HAR using wearable sensors*, the authors in [2] use convolutional neural networks (CNN) on sequential data of multiple inertial measurement units (IMU). Three IMUs are worn by workers on both wrists and the torso. This way, an order picking process in warehouses can be analyzed, fusing data from all sensors to classify relevant human activities such as walking, searching, picking and scanning. In a similar fashion [15] apply different on-body sensors (RFID, force-sensitive resistor (FSR) strap, IMU) in a "motion jacket" worn by workers, and environmental sensors (magnetic switches and FSR sensors) to detect activities in the automotive industry, such as inserting a lamp, mounting a bar using screws and screwdriver, and verifying the lamp's adjustment. We decided to use contactless activity detection and avoid the instrumentation of workers, because we expect limited user acceptance when wearing sensor equip-



**Fig. 1.** Flow chart representing the activity detection process within one work step.

ment. In addition, using wearables the reloading of batteries and mechanical signs of fatigue during usage may be issues the operator wants to avoid. Thus, an easy and light set-up is suggested applying state-of-the-art AI technology to the problem of workflow tracking in manual assembly enabling conformance checking and optimization.

### 3 Concept

BPM has a high potential to support the implementation and adaption of process models which control and sense the assembly workflow. We provide a concept showing how events from the shop floor are correlated to tasks in the model, allowing the supervision of time and quality constraints defined ex-ante in the form of reference times and material states. In addition, the collection of timing information and error frequency supports the optimization of assembly workflows, e.g. through conformance checking. To enable this kind of supervision and optimization, the detection of critical activities within one work step is necessary.

#### 3.1 Activity Detection

Figure 1 shows the three-tiered activity detection: first, grasps to container boxes are detected to analyze which part the worker assembles next (*Grasp Detection*). After performing the actual assembly step, the worker places his hands next to the workpiece, where they are detected by the *Hand Detection* module. At this point in time, an image is captured, as there cannot be any occlusions or motion blur. This image is then used for *Material Detection*, analyzing which workstep the assembly led to, i.e. if the step was correct and the next assembly step can be taken or whether one of several possible failure states is reached and a recovery strategy needs to be applied.

**Grasp Detection** involves the appearance of a foreground object (user's hand) over a stationary background (formed by the image of the container box) seen by the camera observing the container box from the top. It is assumed that this kind of activity occurs when a part is removed from its designated container box. Therefore, activity zones have to be marked within the image to detect the worker's hand. In the process model, material grasping and grasp detection form the first step after receiving the instruction for the current assembly task. Such a system facilitates the supervision of assembly workstations equipped with a

lot of similar-looking materials without the effort of hard-wiring the system with the workstation, as in case of classical pick-supervision.

**Hand Detection** allows the active confirmation of work steps by the worker without reaching to distant touch screens or buttons. With correct and complete execution, this enables positive feedback and forms the point in time when image data for material detection is gathered. This approach ensures that no hands occlude the workpiece and that the workpiece does not move, thereby avoiding motion blur. For the worker, this procedure offers more safety, since subsequent work steps are only started after successful confirmation of the previous ones. In the case of a mistake, the operator is supported with fine-grained assistance.

**Material Detection and State Classification** is then used on the acquired image to confirm whether the material parts being grasped are correctly assembled in the desired way. This involves verifying if each part is correctly located in the expected orientation. A camera is used to monitor the assembly region (where assembly of product is being performed) which facilitates verification and quality assurance based on the image information. Each material state in the process model, including all error states, is considered as a separate class. For each of these classes, a set of images is acquired to train a classifier. Whenever an image is captured after hand detection, this image is classified to the corresponding state in the BPM, which allows jumping to the next step in the worker guidance system (WGS), a system showing instructions about the current assembly task on a screen, or intervening in case of error states. The training images can be acquired either in a separate gathering process (as done for the presented evaluation), or confirmations/correction input to the WGS can be used to label images with corresponding classes, thus supporting cost-neutral data capture in the wild.

### 3.2 Process Model

The process model controls the assembly workflow, communicating with activity detection software and the worker guidance system. The model contains the logic and is configured with a set of properties in a graphical BPMN editor, which allows the coupling of events to the model without the specification of concrete devices and due to the generic implementation without further software implementation effort. The binding between model and components in the assembly workflow is based on a topology describing the value range of the property variables necessary to run a concrete workflow.

Messages are divided into *instructions* published by the model controlling software components and *activities* published by software components detecting material picking and placing or user input from the WGS. Every message contains the business key identifying the process instance and a time stamp. Instructions are subdivided according to the destination (*WGS/ActivityDetection*).

A *WGS instruction* contains the mandatory property work step index (used to gather descriptive information and media to explain and present the current assembly task to the user), a list of expected materials, and a list of expected orientations. Optionally, an error is set when the list of found materials or found orientations does not reflect expected ones, e.g. when *WrongMaterial* or *WrongOrientation* occurs. Then, a correction instruction is presented on the WGS.

An *ActivityDetection instruction* contains the expected materials, orientations and the assembly activity, e.g.,  $grasp_t$ ,  $insert_t$ , or  $rest_t$ , where  $t$  is the reference time for one activity. The time  $t$  can be used to measure the deviation from the manufacturing time planned. To control software components, we introduce the action state that contains the target state the respective component should occupy, e.g. *start* or *stop* the detection of an activity.

An *ActivityDetection event* is sent when the activity detection was started by the controlling process instance and an activity was detected. Then, the activity detection module delivers the properties material (expected & detected), orientation (expected & detected), activity and a type, which indicates if the source of the detection was from an *automatic* (activity detection) or *manual* (confirm button) origin. Since activity detection might fail, as a fallback it is always possible to confirm an activity by pressing a button on the WGS screen.



**Fig. 2.** Assembly workstation equipped with 1 touch screen, 2 cameras and 6 light sources (middle); example material states and 7 light scenes for one part (left); grasp detection using background subtraction with the ‘U’-shaped activity zones (right).

## 4 Implementation

The implementation of the concept presented in the previous section implies an example assembly workflow set up in the lab (Subsect. 4.1), an implemented BPMN 2.0 process model controlling and tracking this workflow (Subsect. 4.2), and three activity detection modules detecting material grasping, hands, and material positioning (Subsects. 4.3–4.5).

## 4.1 Assembly Workflow and Apparatus

To test and iteratively improve our system, we designed an assembly workflow consisting of four assembly steps. The product to be assembled consists of three printed circuit boards (PCB) to be connected and placed in one 3D printed case. The four materials are provided in small load carriers (SLC), boxes common in manufacturing when dealing with small parts. In steps 1 to 3, material is removed from the SLCs and assembled in the work area on top of the workbench in front of the worker. In step 1 the case is removed and placed with the open side up in the work area. In step 2 one PCB is removed and inserted into the case. In step 3 the two remaining PCBs are removed, connected and inserted into the case. The removal of two parts in parallel is common practice to improve the efficiency in assembly workflows. Finally, within step 4 the assembled construct is removed from the work area and inserted into a slide heading to the back of the assembly workstation. The whole workflow is supported by a basic worker guidance system (WGS) running on a touch screen, showing textual instructions and photos of the relevant materials and of the target state in the respective assembly step.

The assembly workstation, shown in Fig. 2, is made from cardboard prototyping material and instrumented with two consumer-electronics RGB cameras: (1) Logitech C920 HD Pro and (2) Logitech BRIO 4K Ultra HD. The first camera is mounted on top of the assembly station and pointed at the four SLCs loaded with material. The second camera is mounted to the shelf carrying the SLCs and is aimed at the work area.

## 4.2 Process Model

A process model, modeled in the BPMN language and shown in Fig. 3, controls the activity detection modules and the WGS. It consists of service, send and receive tasks. Service tasks initialize a new work step and can be used to set the relevant parameters necessary for the subsequent work step. Here, we set the index  $i$  of the step. Send tasks control components necessary to execute the work step and generate *instruction* messages sent to a target component  $c$ . Receive tasks wait for acknowledge events confirming an *activity*  $a$  in the current work step.

Send tasks that initialize the activity detection for one concrete material from graph 1 trigger the initialization of the process modeled in graph 2. There, per material, the grasp, hand and material detection are started and stopped in successive order. Within a send task of type start, the activity to observe is defined. When an activity occurs two options exist: (1) the correlated event occurs as expected and the activity is stopped by the subsequent send task, or (2) a correlated activity occurs that detects a behavior that does not match the expected state. Then, the exception handling is started, instructing the worker to correct his activity. Finally, when the process modeled in graph 2 ends, the receive task in graph 1 is informed and continues.

Error handling is important and allows the system to intervene when states are detected that have been classified as erroneous. 38 classes including error states have been generated with the existing material parts by rotating parts on their positions. The BPMN model was re-designed to instruct the worker during the arrangement of materials such that images can be taken in the training phase of the system covering a variety of lighting conditions.

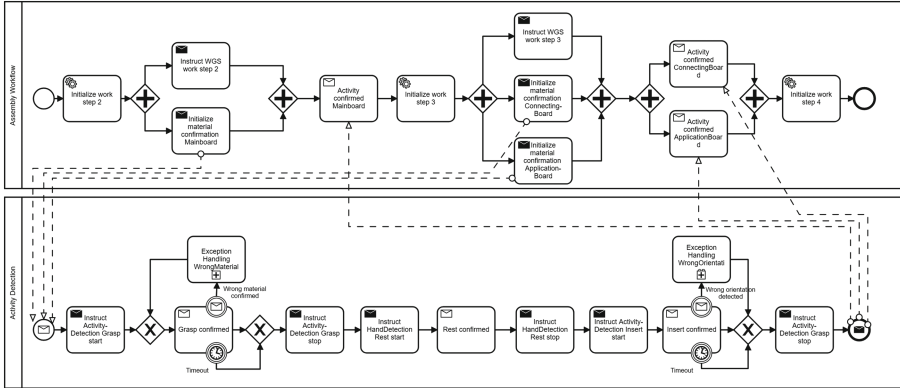


Fig. 3. Assembly workflow snippet (top) and activity detection process (bottom) modeled in BPMN.

### 4.3 Grasp Detection

The activity zones, shown in Fig. 2, are marked outside the interior of the container boxes because a background subtraction algorithm adapts to a changing background, e.g. when a part is removed from its container box. The activity zones are ‘U’-shaped since the direction of approach of the hand is not always perpendicular to the breadth of the container box. The red regions are marked during system setup and the yellow activity zones are automatically identified as a corollary.

The procedure of detecting the start and end of a grasping activity is as follows: When the number of foreground pixels exceeds the total number of pixels in the activity zone by a certain user-defined percentage (40%), activity is said to be detected. To filter noisy indications, the start and end of an activity is detected once there is a considerable number of video frames. Hence a recent history of frames is always observed: Only when a certain user-defined percentage of frames in the recent history of frames contain activity/inactivity is grasp start/end indicated.



## 4.4 Hand Detection

After the user has assembled the grasped material, he or she actively confirms the work step by putting his hands on designated areas of the workstation close to the assembly area, which have electrically conductive metal plates integrated into the assembly worktop. Those plates are connected to a capacitive sensor. The sensor measures the capacitance of the capacitor, which is formed by the electrodes “metal plate” and “operator”. By laying down both hands, the measured capacity reaches a characteristic value, which initializes the manual confirmation of the operator’s work step and forms the point in time when the photo for material detection is taken. As these metal plates are integrated directly next to the assembly area, there is no need to reach distant displays or buttons.

## 4.5 Material Detection and State Classification

As described in Sect. 3, to detect whether materials are assembled correctly, *State Classification* has been implemented based on the image captured at hand detection. Images for every state of the BPM are gathered and a classifier is trained, predicting the state in the BPM based on the image captured during hand detection. Vast improvements in image classification results using deep convolutional neural networks have been achieved in recent years; therefore, we decided to also use convolutional neural nets for our task. Since our task is comparatively simple, we do not apply complex and extremely deep convolutional networks like ResNet [3], but instead design our own very simple network: we use 5 (convolutional, convolutional, max pooling) blocks with filter size of 3 by 3, pool size of 2 by 2, and relu activation, followed by a dense layer of 512 (relu activation), and the final classification dense layer performing softmax. Dropout is used to reduce overfitting.

To avoid long data gathering phases, we perform an initial training procedure based on parts of the ImageNet dataset [14]. For this, we downloaded a total of 419 classes of the set, including 178 classes we considered roughly related to PCB assembly, such as ‘electrical circuit’, ‘printed circuit’, and ‘circuit board’. We resize all images to  $224 \times 224$  pixels and train the network for 300 epochs using a batch size of 64, RMSProp optimization, and a learning rate of 0.0001. These weights are stored and used as a basis for the training process on the images specific to our assembly task. This pre-training should reduce the amount of images required, as basic features like edge, shape, and color detection can already be learned from the ImageNet data. The dense and classification layers at the end are randomly initialized when fine-training on our dataset, as these can be considered specific to the ImageNet data.

## 5 Evaluation

We conduct an evaluation, testing the individual parts of the activity detection module and analyzing to what extent these can be used to automatically gather insights into the assembly process to optimize it in later steps.

## 5.1 Grasp and Hand Detection

The grasp and hand detection is evaluated using a series of experimental runs involving 6 users (1 *f*, 5 *m*) with different hand sizes (circumference:  $\bar{x} = 21.58$ ,  $\sigma = 1.64$  cm; length:  $\bar{x} = 18.83$ ,  $\sigma = 1.77$  cm; span:  $\bar{x} = 21.67$ ,  $\sigma = 2.29$  cm). During the experiment, the user grasps a part, places it on the worktable and confirms by resting the hands. The user then returns the part and confirms by resting the hands. This procedure is repeated for each part and box with right and left hand alternating 2 times per part in four repetitions comprising all parts. This leads to a 2 (hands) \* 2 (remove and return part) \* 4 (parts) \* 4 (repetitions) = 64 grasp start, stop and rest events. The activity start and stop are monitored by a supervisor with an annotating application to generate ground truth information. The rest detection failed only in 2 cases where two users curved their palms, leading to no sensor response. In total, an accuracy of 99.22% was achieved. For grasp detection, 2 pairs of start and stop events failed where one user approached from angles where the activity zone is least disturbed, such that the threshold of 40% foreground pixels was not reached in the activity zone. In total, an accuracy of 99.48% was achieved.

## 5.2 Material Detection and State Classification

Training of a neural network usually requires manually labeling a large amount of data samples. For the system described in this paper, we used the BPMN model to automatically label the assembly steps: a custom model which, in combination with the worker guidance system, not only instructed the participant to perform the usual assembly steps, but also directed her to generate erroneous states (i.e. wrong part placement or orientation). This made it possible to semi-automatically (the participant still had to place her hands next to the workpiece after each step) take a picture of every state of the BPMN-model and label it at the same time. Since different lighting conditions easily occur in a real life setting, we also took these into account during the training data acquisition.

To simulate various lighting condition, six Philips Hue lights were placed in different positions around and on the assembly workstation. Using the Philips Hue API allowed us to pragmatically change the lighting scenarios, which results in different shadows on the pictures of the workpieces. During data gathering, the remaining lighting conditions were kept stable (shutter closed, room and workstation lights on). We used seven different simulated lighting conditions (including all lights off, i.e. room lighting) per assembly state. The rationale behind this was to allow for different lighting preferences of the individual workers. Each iteration consisted of 19 states, generating 19 \* 7 images and with a duration of approximately 8 min (25 s per assembly step). Two types of iterations were considered, orienting all parts in two directions (left and right) leading to 38 classes in total.

With this setting, we were able to generate 3990 images in 30 runs within 4 h. To avoid irrelevant parts of the surroundings, such as hands or tools, being present in the image, the camera has to be fixed. This allows focusing on the

rectangular area that contains the assembled parts which are relevant for classification and detection, which is a requirement to run our system. Overall the goal is to quantify how well the approach works to determine whether material was assembled correctly. For this, we use 50% of the labeled image data for training, 25% for validation, and 25% for testing.

As described in Sect. 4.5, the neural network used to classify the various assembly states was pre-trained on a subset of ImageNet, and then all except for the dense layers are fine-tuned on the training data specific to our task. For this, we again resize all input images to 224 by 224 pixels and train for 25 epochs using a batch size of 8. As expected due to the simple nature of the problem (in comparison to competitions like the ImageNet Large Scale Visual Recognition Challenge (ILSVRC)), the classification performance is very good: a test accuracy of 99.25% was achieved on this 38-class problem.

## 6 Conclusion

Methods from AI, such as computer vision and machine learning, were tailored to an Industry 4.0 use case and may increase an organization's competitiveness through awareness of error rates and timepass enabling backtracking and intermediate intervention. We have shown that an easy-to-set-up tool set of 2 cameras, 1 capacitive sensor, 6 light sources and activity detection software trained within 4 h has the ability to generate accurate sensor events. Viewing these data through a "process lens," the measurement of time and quality in manual assembly workflows, and thus the optimization of processes, is enabled.

Since it was shown how an Industry 4.0 organization can keep track of its manual assembly workflows, it will be interesting to investigate the potential of unsupervised methods in combination with our approach to discover workflows automatically towards a novel concept of 'video-to-model'. New methods to discover assembly workflows in planning and construction phases, and to monitor or check their conformance online and offline, will provide valuable input for process mining.

**Acknowledgments.** This research was funded in part by the German Federal Ministry of Education and Research (BMBF) under grant number 01IS16022E (project BaSys4.0). The responsibility for this publication lies with the authors.

## References

1. Cavanillas, J.M., Curry, E., Wahlster, W. (eds.): *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe*. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-21569-3>
2. Grzeszick, R., et al.: Deep neural network based human activity recognition for the order picking process. In: *Proceedings of the 4th International Workshop on Sensor-based Activity Recognition and Interaction*. iWOAR 2017, pp. 14:1–14:6. ACM Rostock (2017)

3. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
4. Hull, R., Motahari Nezhad, H.R.: Rethinking BPM in a cognitive world: transforming how we learn and perform business processes. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 3–19. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-45348-4\\_1](https://doi.org/10.1007/978-3-319-45348-4_1)
5. Jaroucheh, Z., Liu, X., Smith, S.: Recognize contextual situation in pervasive environments using process mining techniques. *J. Ambient Intell. Humaniz. Comput.* **2**(1), 53–69 (2011)
6. Kagermann, H., et al.: Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry; Final Report of the Industrie 4.0 Working Group. Forschungsunion (2013)
7. Knoch, S., et al.: Automatic capturing and analysis of manual manufacturing processes with minimal setup effort. In: International Joint Conference on Pervasive and Ubiquitous Computing. UbiComp, pp. 305–308. ACM, Heidelberg, September 2016
8. Knoch, S., Ponpathirkoottam, S., Fettke, P., Loos, P.: Technology-enhanced process elicitation of worker activities in manufacturing. In: Teniente, E., Weidlich, M. (eds.) BPM 2017. LNBIP, vol. 308, pp. 273–284. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-74030-0\\_20](https://doi.org/10.1007/978-3-319-74030-0_20)
9. Lasi, H., et al.: Industrie 4.0. *Wirtschaftsinformatik* **56**(4), 261–264 (2014)
10. Lenz, C., et al.: Human workflow analysis using 3D occupancy grid hand tracking in a human-robot collaboration scenario. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3375–3380, September 2011
11. Marrella, A., Mecella, M.: Cognitive business process management for adaptive cyber-physical processes. In: Teniente, E., Weidlich, M. (eds.) BPM 2017. LNBIP, vol. 308, pp. 429–439. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-74030-0\\_33](https://doi.org/10.1007/978-3-319-74030-0_33)
12. Poppe, R.: A survey on vision-based human action recognition. *Image Vis. Comput.* **28**(6), 976–990 (2010)
13. Roitberg, A., et al.: Multimodal human activity recognition for industrial manufacturing processes in robotic workcells. In: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction. ICMI 2015, pp. 259–266. ACM, Seattle (2015)
14. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis. (IJCV)* **115**(3), 211–252 (2015)
15. Stiefmeier, T., et al.: Wearable activity tracking in car manufacturing. *IEEE Pervasive Comput.* **7**(2), 42–50 (2008)
16. Thoben, K.-D., Pöppelbuß, J., Wellsandt, S., Teucke, M., Werthmann, D.: Considerations on a lifecycle model for cyber-physical system platforms. In: Grabot, B., Vallespir, B., Gomes, S., Bouras, A., Kiritsis, D. (eds.) APMS 2014, Part I. IAICT, vol. 438, pp. 85–92. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-662-44739-0\\_11](https://doi.org/10.1007/978-3-662-44739-0_11)