Teach Me How! Interactive Assembly Instructions Using Demonstration and In-Situ Projection

Markus Funk, Lars Lischke, Sven Mayer, Alireza Sahami Shirazi and Albrecht Schmidt

1 Introduction

In industrial settings production effectiveness and efficiency is paramount. Over the last 50 years automation and robotics massively changed how goods are manufactured. It is foreseen that over the next decade a further revolutionary shift to more flexible production systems will happen, as outlined in the smart factory [35] and Industry 4.0 [27] initiatives. However in most domains production is not fully automated and human workers still play an essential role. For example in the car industry, human workers are cooperating with robots in complex assembly processes. With individualized products many variants are produced in the same production line at the same time. Also, as storage costs are increasing, ordered products are produced on demand—just when they were ordered. This process is called lean manufacturing. However, in such flexible production environments where many different variants of a product are assembled, the task of the worker becomes more and more complex. Humans are creative and have great skill for manipulating objects. However dealing with large number of variants is cognitively demanding and typically high level instructions are required (*this screw should be attached to this part*). Low level instructions (e.g. how to hold the screw, how to insert it into a hole, how to hold the screw driver, etc.) are not required as these motor-cognitive tasks are simple for humans (in contrast to a robot) [48]. Workers have to understand which variant they are creating and what steps are required. With small lot sizes and frequent changes, classical training and teaching approaches do not scale. Neither learning all possible variants upfront, nor getting a traditional training session each time the product on the assembly line changes is a viable option. The method of choice is to provide the information required for the production when the worker needs them.

The majority of the work has been conducted while he was a researcher at the University of Stuttgart.

M. Funk (✉) · L. Lischke · S. Mayer · A. Schmidt
University of Stuttgart, Pfaffenwaldring 5a, 70569 Stuttgart, Germany
e-mail: makufunk@hotmail.com

A.S. Shirazi
Yahoo Inc., 701 1st Ave, Sunnyvale, CA 94089, USA

J. Huber et al. (eds.), *Assistive Augmentation*, Cognitive Science and Technology, https://doi.org/10.1007/978-981-10-6404-3_4

In traditional production with large lot sizes and a small number of variants it was useful and cost-effective to create training and information material upfront. Depending on the task and environment assembly manuals were created as paper-based instructions or videos. More recently assembly instructions were also created for in-situ systems, e.g. Pick-by-Light [21] or Augmented Reality (AR) systems [3, 8, 42]. The cost for creating instructions can be divided by the number of products created based on this instruction. Consider the following example of assembling a refurbished starter for a car (similar to the one used in the study). The average assembly time for the product by a worker is about 3 min. If we assume the creation of a traditional tutorial video, this will take 120 min, creating written instructions takes 60 min, and for a set of instruction based on demonstration we estimate 6 min. For a lot size of 10.000 the cost of creating the instruction is less of an issue as the assembly time will be the major cost factor. However for a lot size of 20 it is clear that the creation of instructions becomes a major issue. Skill acquisition for individuals and skill transfer within the workforce becomes more important and a major factor for competitiveness in flexible production environments. In our research, we envision that skills of workers can be captured with little or no effort and can be transferred to others to pick them up with little effort. Continuing the example from above, we assume that for the starter with the lot size of 20 a skilled worker would do one assembly to remind herself of the best way of doing it, then she would assemble a second one where the system is used to record the assembly, and then the remaining 18 starters could be assembled by untrained workers. In this chapter, we empirically compare two approaches for recording and using of the instructional material: videos and interactive assembly instructions, which are semantically rich and where the information is embedded and presented step by step.

Extending our previous system [16, 17], we have developed a functional system that automatically generates these interactive assembly instructions using the Programming by Demonstration (PbD) approach (Fig. 1). While the user demonstrates

Fig. 1 The system provides visual instructions for supporting workers during the assembly of an engine starter. It highlights the position where a part should be assembled and checks if it is assembled correctly. Instructions were created using a simple programming by demonstration approach

an assembly task by assembling the parts step by step, our system detects the currently obtained part and the position where it is assembled. Using this information, the system automatically creates an assembly instruction, where the semantics of each step is retained. With this PbD approach, interactive in-situ instructions can be created almost as fast as recording a video of assembling the product but still retain all features of interactive instructions. Our approach enables the instructor to physically show a new workflow to the system, same as it would be shown in front of a camera or to a new worker. Further, the system can use the recorded information to provide step-wise instructions using in-situ projection. It highlights the bin where a part should be picked from and the position where it should be assembled in each step. Therefore, our system provides a new means for process engineers for creating interactive instructions and a new way for workers to use assembly instructions. We believe that this work adds to the area of assistive augmentation by introducing a stationary assistive system that provides cognitive assistance during assembly tasks.

The contribution of this chapter is threefold: first, we present a system that automatically detects work steps and creates a semantically rich assembly instruction while an assembly is performed using a depth camera. The system also uses in-situ projection to provide the steps for assembling a product. Second, we compare video based instructions and Augmented Reality-based step-by-step in-situ projection. With 32 participants using reproducible tasks of different complexity, we compare the impact of the different representations for the quality and performance of the assembly. The study shows that, especially for more complex assemblies, the error rate (ER) decreases, the assembly is faster, and the mental demand is reduced using in-situ instructions. Third, we investigate the effort required for creating instructions in a realistic work environment with industrial workers. We present the findings of a user study with expert users comparing three approaches for creating assembly instructions: traditional video recording, using a graphical editor, and automated extraction of instruction using the described systems. The results show that using our system, assembly instructions can be created faster with less perceived cognitive load in comparison to using a graphical editor while the effort is comparable to traditional video recording. We validate the created instructions in an industrial setting with 51 workers in a car production plant.

The chapter is structured as follows: after reviewing the prior work, we present an interactive assembly system that creates semantically rich assembly instructions from a demonstration. We describe a laboratory study in which we compare video based instruction and the in-situ projection. Then two studies in an industrial setting, one for creating instructions and one for using instructions, are reported. The participants in the studies are skilled workers for creating the instructions and unskilled workers for using the created instructions. As a task, we use the assembly of a refurbished car starter. Finally, we finish the chapter with discussing implications.

2 Related Work

Creating and providing assembly instructions using interactive systems has been the subject of various research. In the following, we provide an overview in relevant research areas for creating and presenting interactive assembly instructions, namely, Programming by Demonstration, projected surfaces, and Augmented Reality.

2.1 Programming by Demonstration

PbD (also referred to as programming by example) was initially proposed to enable users to record macros without knowing any programming language or writing code. This approach has been adopted by many application domains which comprise desktop applications like MS Excel, computer-aided design, and text editing [33]. Thereby, a user's actions are translated into a textual procedure, which later can be played back and altered. For example, the Peridot system [37] enables interface designers to demonstrate how a UI should look like rather than having to program it. Recently, Kubitza and Schmidt [31] introduced a framework that enables non-programmers to use PbD to program for smart environments.

The PbD approach is also used to teach new motion sequences to humanoid robots by recording movements of a human worker. Aleotti et al. [1] reproduce and optimize measured trajectories of a human worker. The trajectories can then be used to infer high level actions [6]. After defining actions, the sequence of the actions can be played back and altered. Instead of programming physical robots, Marinos et al. [36] use a PbD approach to rapidly create animations for a virtual robot inside a blue or green box of a virtual studio.

Overall, previous work shows that even non-programmers can use a PbD approach for creating digital content, programming physical robots, and defining procedures. This rapid creation of digital content does not need special training as the actions that are performed by the users are natural actions that users would also do without using a computer. In our system, combining the PbD approach with interactive surfaces and AR, we enable users to create interactive projected instructions for humans.

2.2 In-Situ Projection and Interactive Surfaces

Projecting information directly into the interaction space or onto objects has been used to augment real world objects with digital information or to display information in-situ. Pinhanez [38] uses a rotating mirror to create displays out of arbitrary surfaces and to augment objects with information. Combining this technology with a camera, projected surfaces become interactive. In the Touchlight system [44], Wilson uses two RGB-cameras and computer vision techniques to detect touch input

on a projected surface. The LuminAR [34] system integrates such a camera-projector system into an anglepoise lamp. The lamp can project information next to a recognized object on a desk. Furthermore, it can detect performed gestures. On the other hand, other projects applied in-situ projection to different areas, e.g. the kitchen [32] or sterile training areas [39].

With the proliferation of depth cameras, sensing interaction on projected surfaces has become easier. Wilson [45] suggests an algorithm that enables sensing of multi-touch without using an RGB image. This algorithm was improved and provided as a framework in the Ubi Displays toolkit [24]. With this toolkit a user can define multiple touch-enabled areas that have their own projected content. The dSensingNI project [28] combines Wilson's algorithm with gestural user inputs in their tabletop system. Furthermore, they support detecting the presence, volume and orientation of cubical objects using a top-mounted Kinect. Although dSensingNI is capable of detecting stacked objects, the system is not able to detect if a construction is correctly assembled.

Overall, related work showed how to augment physical objects with digital information using in-situ projection. Further, user interaction on these projected displays can be detected using RGB or depth cameras. Our system also uses top projection to provide in-situ information. Additionally, it can detect if a construction is assembled correctly using a depth camera and computer vision.

2.3 Providing Assembly Instructions for Training Workers

Videotaping of a manual assembly process is a straightforward approach for creating assembly instructions, which is used to teach assembly procedures to untrained workers. These so-called Utility Videos (e.g. Memex[1]) are produced by professional companies for training unskilled or new workers in a new assembly task. On the other hand, systems providing interactive AR instructions [11] have been suggested to assist workers during assembly tasks. For example, Pick-by-Light systems visually show the worker, where the next part has to be picked from [3], or how a part has to be assembled [4]. Also Head-Mounted Displays (HMDs) can show the worker the next part and where the part has to be assembled [10, 23, 42]. More recently, assistance technologies focused on projecting instructions directly into the workers field of view (e.g. Light Guide Systems[2]). This in-situ projection reduces the complexity of the given feedback, as it is projected directly into the work space, instead of giving feedback on an external monitor. Such projected instructions are usually created using a graphical editor. However, with frequently changing variants of the same product, creating and maintaining instructions is cumbersome. Instead of being able to just alter the changed steps of the variant's workflow, the instructor often needs to change the whole workflow as even small changes effect succeeding work steps.

[1] http://memex-academy.eu/ (last access 03-18-2016).

[2] http://www.ops-solutions.com/ (last access 03-18-2016).

2.4 Augmented Reality in Assembly

Industrial Augmented Reality (IAR) is now almost always present in a manufactured product's life-cycle. Experiencing a designed product can be done immediately [13], industrial robots cooperating with human workers can be programmed using AR-debugging approaches [12], order picking can be supported using HMDs [22] or projector carts [21], and maintaining existing machines and products can be supported directly on site [46]. Workers can even be motivated during the work tasks by using IAR for gamification [30].

Prior work has used AR to provide assembly instructions. An overview about this topic is presented by Büttner et al. [9]. A strand of work has augmented parts of a product with sensors. Antifakos et al. [2] use instrumented tools and assembly parts to infer a user's current action and suggest proactive instructions for assembling an IKEA PAX wardrobe. Compared to a printed manual, their system can dynamically react upon a user's action as it is aware of all possible assembly orders rather than printing one fixed order. However, integrated sensors may influence the design of the product.

Instead of augmenting the assembly parts, other research proposed mobile systems for displaying interactive assembly instructions by augmenting the users with sensors. For example, Ward et al. [43] equip the user with body worn microphones and accelerometers to infer the user's current activity in an assembly environment. Even when combining multiple features [7] to recognize an activity more reliably, a body worn system unfortunately cannot detect if a part is assembled correctly.

Using HMDs is another approach that has been explored to display assembly instructions during work tasks [11]. It has been shown that it can reduce the task completion time (TCT) and mental workload [42]. This concept has been adapted to several domains. Through a user study, Henderson et al. [26] report that users have less head movements using HMD-based AR instructions while repairing a vehicle. Zauner et al. [47] use AR markers to provide assembly instructions on a HMD for assembling furniture. Salonen et al. [40] are also using a marker approach while experimenting control modalities. However, overall the feedback on these assembly instruction systems has to be explicitly advanced to the next work step.

While the aforementioned approaches are for mobile settings, assisting systems for stationary setups have been explored, too. For example, Bannat et al. [3] present a framework using a top-mounted RGB camera to detect bins automatically based on their color and shape. Once the position of the bins in known, their system uses the RGB camera to detect the position of the worker's hand. In their system assembly instructions are shown on a monitor close to the work area. The system highlights the next bins to pick from using a top-mounted projector. Korn et al. [29] extends this approach by using a top-mounted depth camera instead of a RGB camera and a top-mounted projector in production environments. The position of the bins and the position of an assembled part have to be defined manually using a graphical editor. Their system then highlights the bin to pick from. As their system cannot automatically detect the correct assembly in each step, it uses projected buttons that the

user can manually advance the projection to next step. Recently, Funk et al. [16, 17] investigated the potentials of using in-situ projected instructions at the workplace to support workers with cognitive impairments. They found that using in-situ projected instructions workers with cognitive impairments can assemble more complex products without increasing the time or errors per work step. Further, they found that using a simple contour-based highlighting as assembly instruction is perceived as better than video, pictorial, and no instructions [19].

Overall, previous work suggests using a setup consisting of a top-mounted projector and a depth camera to display instructions using in-situ projection. In our system, we also use that setup to detect picking from bins. We additionally use the depth camera to detect if the assembly is performed correctly. In contrast to prior work, our system automatically creates in-situ feedback based on a demonstrated assembly and automatically highlights the bin to pick from. Overall, by using PbD, our system requires no additional effort from the user when creating instructions except assembling the product.

3 Instructions Creation Through Demonstration

We developed an interactive system for creating and providing semantically-rich assembly instructions, which uses the concept of PbD to create instructions. Hereby, the system is able to automatically create instructions for an assembly task while it is being performed. It detects out of which bin a part is picked and where the part is assembled. During the assembly, the system can project assembly instructions directly into the work area. Accordingly, it highlights which bin to pick a part from, and at which place it should be assembled on the workpiece carrier. In the following, we give an overview about hardware and software of the system, which is an extension of the software presented in [16].

3.1 Hardware Setup

We designed our system that it reflects an assembly workplace found in the industry. Figure 2 shows the system and its components. It consists of a top-mounted projector, a Kinect depth sensor, a number of bins, and a workpiece carrier. The bins in the back of the system (Fig. 2c) contain the assembly parts. Tools needed for the assembly task are placed at the side of the system (Fig. 2e). The steel plate in Fig. 2d is a workpiece carrier that holds parts during the assembly. In an industry setup, workpiece carriers are exchanged between work places using a skate wheel conveyor and then are fixed from below using a pneumatic clasp. We firmly mounted the workpiece carrier on the table to prevent it from moving while conducting work steps.

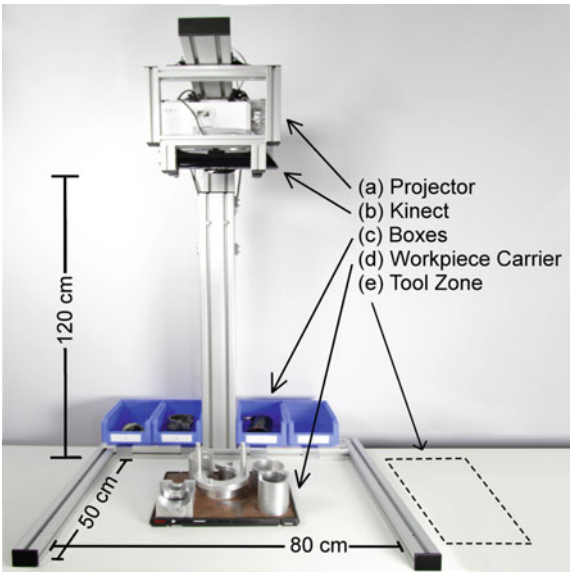

Fig. 2 Our system consists of four components: **a** top-mounted LED-projector, **b** kinect for windows, **c** bins containing the assembly parts, and **d** the workpiece carrier holding the product. The area in **e** depicts the tool zone

To make our system transportable, we built an aluminum frame which holds the Kinect, the projector, and the bins. The projector highlights the bin from which the user should pick a part from, tools that should be used, and the position on the workpiece carrier where the part has to be assembled. The Kinect detects if a part was picked from a bin, if a part is assembled correctly, and if a tool is used. The number of bins, the content of the bins, and the workpiece carrier change according to the manufactured product and the steps that are performed at the work place. With our current setup, the system can handle a maximum of eight bins (2 rows $\times$ 4 bins) due to the limited angle of the Kinect which has to cover both work area and the bins. Our system provides a predefined layout of the bins and a predefined area for putting tools on the right side of the system. The layout of the bins and the area for putting tools can be changed and customized using a graphical interface.

In our setup, we use an Acer K335 LED-projector with 1000 ANSI Lumen and a Kinect for Windows running on a depth resolution of 320×240 pixels with 30 frames per second. Both Kinect and projector are mounted 120 cm above the surface and are facing the table. They are calibrated using the 4-point calibration of the Ubi Displays toolkit [24].

3.2 Work Step Detection

For detecting assembly steps and creating instructions, we define a high-level representation of performed actions (c.f. [6]). We call this high-level representation a

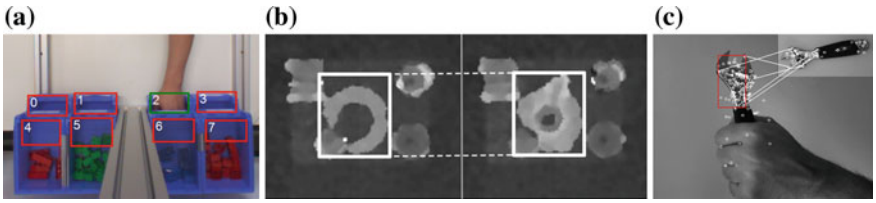

Fig. 3 Overview about the triggers used in our concept: **a** detecting a hand entering a bin, **b** detect object placement based on depth data, **c** recognizing the presence and the absence of an object compared to a previously taken reference image using computer vision

workflow, which consists of a finite number of work steps. Each work step has an initial state and a trigger condition (trigger) for advancing to the next step. A trigger is activated by one of the following three actions: (1) pick a part from a bin, (2) assemble a part that was just picked, (3) use a tool on the product.

Using the trigger concept, actions in each step of the assembly can be detected. Further it enables implicit interaction [41] with the system to trigger the next step of the workflow. In our model, we define three triggers which notify the system that one of the actions was performed (see Fig. 3).

3.2.1 Pick Detection

Our system uses the top-mounted Kinect to detect when a hand enters a bin. The placement of the bins can be defined using a graphical interface where the user can adjust the size and the position of the bin directly in the Kinect's RGB image (Fig. 3a). Once the position of the bins is defined, the system stores a depth map of the bin's area to continuously compare it to the most recent depth image. Using this technique, we can define 3D cubes in the work area that layover the bins. When the system registered a pick from a bin, the bin is briefly highlighted by the projector. To be robust against depth sensor noise, we consider at least 4 mm changes in depth value. Our algorithm compares each depth pixel inside the cube to the previously stored state. If the participant picks a part from a bin, the percentage of changed depth pixels becomes larger. If the percentage exceeds a threshold, the bin is triggered. The threshold is dependent on the size of the worker's hand, the size of the bin, and the distance of the camera to the bins. In an informal test we found that a threshold of 63% is a good value to reliably detect the hands of 5 different persons.

3.2.2 Assembly Detection

At the beginning of recording a work step, the system captures the initial depth data as an initial state. Then, the worker can start assembling the product. To capture each work step correctly while recording an instruction, the worker needs to step out of

the work area after each assembly step. The system's built-in movement detection converts color frames into a gray-scale image and subtracts each 15th frame from the previous one. If a difference between the images was found, the system knows that the worker is still performing a task. If no movement was detected for the last 1.5 s, the system assumes that the worker's hands are out of the work area and captures the current depth data. This data is compared to the previously captured initial state by transforming both depth arrays into gray-scale images and subtracting them from each other using EmguCV.[3] This algorithm enables the system to detect where a part was assembled (see Fig. 3) and to distinguish between removing and adding parts. If the area changed is larger than a threshold of 150 pixels in total, it is considered to be a valid work step. The threshold of 150 pixels was chosen empirically and provided a robust trade-off between filtering sensor noise and detecting assembly steps. Afterwards, the latest depth data is stored with the work step as a desired state and visual feedback is given to the user.

When playing back a workflow, the depth data from the desired state is continuously compared to the depth data of the current frame by comparing each pixel. If the current depth frame matches the desired state, the work step is considered to be performed correctly and the system proceeds to the next work step.

3.2.3 Detection of Tool Usage

In our prototype, we predefined a tool zone at the right side of the system, which can be changed and customized. Our system continuously scans the depth data of the defined tool zone and checks the changes in the data. In case, the change in depth data is over a threshold of 63%, our algorithm runs the SURF object recognition algorithm [5] to compare the image of the defined zone to the previously recorded reference image of the object (Fig. 3c). If the object cannot be recognized in the picture, it is considered to be taken and the system assumes that the worker is using it. When the user puts the object back at its place, the depth data changes and the system runs the SURF algorithm again. If the object is recognized again, the system triggers that the object was used and displays visual feedback.

3.2.4 Resulting Instructions

As the system is detecting the object that is picked, the assembly which was performed, and the tool which was used, a semantic description of the performed step is stored for each step. In particular the information about which part is picked and which tool is used helps to add flexibility. Using the resulting instructions can be transferred to another production table (or to multiple tables) where there may be different arrangement for parts and tools. Changing the location of a bin with parts (either automatically detected or manually entered into the system) can then be used

[3] http://www.emgu.com/ (last access 03-18-2016).

to change an existing instruction. Assume a bin containing screws is moved from left to right. As the semantic information is available the visual feedback showing the worker where to pick can be moved to the new position accordingly. In contrast, video-based instructions would need to be re-recorded.

3.3 In-Situ Projection for Visual Feedback

In our system, visual feedback for the next work step is created automatically from the user's actions. As the Kinect and the projector are calibrated, the system knows which pixel in the Kinect's image matches which pixel of the projector. For picking objects out of a bin, the system highlights the correct bin with a green light. Once the user picked a part from the bin, the system projects green light at the position where the part has to be put on the workpiece carrier. Thereby, the highlighted position is calculated automatically by comparing the depth data of the initial and the desired state. As suggested by previous work [19], the system calculates a contour visualization. When a part should be removed from the workpiece carrier, the system highlights the contour of the part on the workpiece carrier with a red light. In case a tool should be used, the system highlights the tool's position with a yellow light in the tool zone.

3.4 Playing Back Workflow Instructions

The triggers are also used for playing back a previously recorded workflow. The system plays each step of a workflow in the order it was defined. If the current step is a picking-task from a bin, the system only advances to the next step if the defined bin is triggered. In case the current task is assembling a part, the system advances when the depth data stored for the desired state matches with the current depth data to an extend of at least 90% accuracy. When the current step is to use a tool, the system checks if the tool is removed from its place. If the tool was absent for at least three seconds and it is put back again, the system considers the object as used and triggers the next step.

4 Study #1: Assembly with Different Complexities

To assess our system using assembly tasks with a different number of steps and complexity, we conducted a user study in our laboratory. Inspired by previous work [15, 16, 18, 42], we decided to use Duplo[4] bricks for creating construction models with

[4] http://www.lego.com/en-us/duplo (last access 03-18-2016).

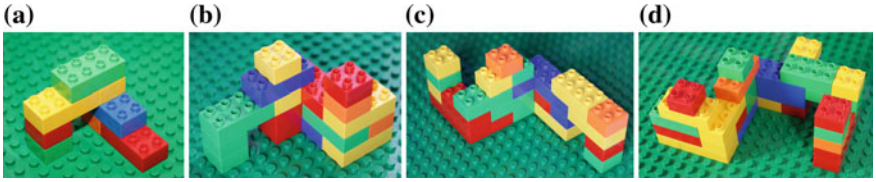

Fig. 4 The constructions used in the lab study with four different complexity levels: **a** 8 bricks, **b** 16 bricks, **c** 24 bricks, and **d** 32 bricks

different numbers of bricks. As the system can monitor a maximum of eight bins, we considered four models with four different numbers of bricks, i.e., 8, 16, 24, and 32. All the four models were created using 8 different types of bricks in five different colors. They all have one arch in the bottom level. Figure 4 shows the four constructions.

4.1 Method

A mixed design was considered for carrying out this study. We used a between-subject design with the type of instruction as the only independent variable with two levels: the video-based approach and the in-situ projection approach. Within the groups, we used a repeated measures design with the number of bricks as independent variable (4 levels). As dependent variables in both groups, we measured the ER, the TCT, and the NASA-Task Load Index (NASA-TLX) score. The order of the repeated measures tasks was counterbalanced according to the Balanced Latin Square.

We created two assembly instructions for each construction model: recording the video, and using the PbD approach. For recording the video instructions, we used a camcorder and videotaped the assembly instructions in HD resolution recorded from over the shoulder of the worker. For recording the projected instructions, we used our PbD system. In both cases one of the researchers performed the assembly task while the instructions were recorded and created. For both conditions, the content of the bins and the bins' arrangement were identical. Each type of brick had a separate bin resulting in 8 different bins.

For the video instruction, a monitor was placed next to the work area (see Fig. 5a). The participant could play and pause the video using the space key on the keyboard at any time during the assembly. For the in-situ projection, the participant sat in the same place in front of the plate and for each step instructions were projected into the work area by either highlighting a bin to pick from or the position a brick should be placed (see Fig. 5b).

The procedure of the study was as follows: after welcoming the participant and giving a brief introduction about assembling products, we collected the demographics. Then, one of the instructions was assigned to the participant. When the participant

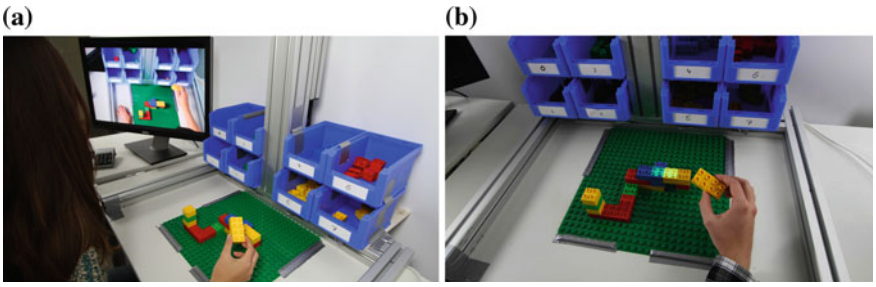

Fig. 5 The setup of the two conditions used in the lab study. The video condition uses a monitor to display the instructions (**a**). The PbD condition projects visual feedback onto the Duplo bricks (**b**)

was ready, the experimenter started the instruction and measured the TCT. The participant was instructed to only use the predominant hand to pick and assemble the bricks as assembling two parts at the same time is not supported by the system. The whole experiment session including hands and the picking from the bins was video recorded for each participant. After assembling each model, the participant was asked to fill in the NASA-TLX questionnaire. The participant repeated this procedure for all four construction models. After the study, two researchers independently watched the videos and counted the errors for each participant. They compared the results and in case of inconsistency, the researchers reviewed the videos together until they came to an agreement.

We recruited 32 participants, 8 female and 24 male with the average age of 25.1 years ($SD = 3.9$) using the University's mailing list. All participants were students in various majors. They had no prior knowledge in assembling the Duplo buildings nor participated in the two previous studies. Furthermore, none of the participants was colorblind. The study was conducted in our lab at the University of Stuttgart.

4.2 Results

We statistically compared the ER, the TCT, and the NASA-TLX score between the four models and the two instruction methods conducting a two-way mixed ANOVA. Mauchly's test indicated that the assumption of sphericity had been violated for ER ($\chi^2(5) = 17.60, p < 0.004$) and TCT ($\chi^2(5) = 23.29, p < 0.001$). Therefore, the degrees of freedom were corrected using Greenhouse-Geisser estimated of sphericity ($\epsilon = 0.73$ for ER and $\epsilon = 0.68$ for TCT). The t-test with Bonferroni correction was considered as post hoc test for all cases.

The analysis revealed that the difference in the ER between the four models was not significant ($F(2.18, 65.36) = 1.94, p > 0.05$). The model with the 24 steps had the largest ER ($M = 0.66$, SD 1.61) followed by the 32-step model ($M = 0.59$, $SD = 1.38$) and 16-step model ($M = 0.47$, $SD = 1.04$). Whereas, the effect on the ER

between the two feedback approaches was statistically significant ($F(1, 30) = 11.20$, $p < 0.002$, $r = 0.39$). The effect size estimated shows a medium and hence substantial effect. The post hoc test showed that the video-based instruction had a significantly larger ER than the in-situ projection instruction ($M = 0.86$, $SD = 1.36$ vs. $M = 0.05$, $SD = 1.36$, $p < 0.002$).

Analyzing the TCT between the constructions showed that it statistically significantly differed ($F(2.05, 61.5) = 217.88$, $p < 0.001$). Post hoc tests revealed a significant difference between all constructions. The 32-step model had the longest TCT ($M = 2.31$ min, $SD = 0.69$) followed by 24-step ($M = 1.83$ min, $SD = 0.70$) and 16-step ($M = 1.10$ min, $SD = 0.31$). Such differences were already expected due to the variation in the number of bricks. On the other hand, feedback approaches had statistically significant effect on the TCT ($F(1, 30) = 63.82$, $p < 0.001$, $r = 0.80$). The effect size indicates a large and substantial effect. Surprisingly, the TCT using the video method took 1.5 times longer than the PbD method ($M = 1.73$ min, $SD = 0.45$ vs. $M = 1.08$ min, $SD = 0.45$).

Furthermore, there was a statistical significant difference in the NASA-TLX score between the constructions ($F(3, 90) = 3.63$, $p < 0.01$). The post hoc tests showed that the difference was only significant between the 8-step and 32-step models ($M = 22.34$, $SD = 16.20$ vs. $M = 27.87$, $SD = 17$, $p < 0.1$). The score between other constructions was not significant (all $p > 0.05$). The average score for the 16-step model was 25.03 ($SD = 17.47$) and for the 24-step construction the score was 26.38 ($SD = 16.93$). The comparison between the methods revealed a statistical significant effect on the mental load ($F(1, 30) = 19.73$, $p < 0.001$, $r = 0.54$). The effect size indicates the effect is large and substantial. The mental load for the in-situ instruction approach was 60% smaller than the video-based instruction ($M = 15.62$, $SD = 25.96$ vs. $M = 35.19$, $SD = 25.96$, respectively).

4.2.1 Impacts of Number of Steps in Assembly

We further assessed the differences between the two feedback approaches for different complexities, i.e. having a different number of assembly steps. To achieve this, for each construction model, we conducted the t-test between the video and in-situ instructions and pair-wise compared the ER, TCT, and NASA-TLX score. The Levene's test conducted in all cases to test the equality of variances. In case the assumption was violated the degrees of freedom were adjusted.

The comparison of ER showed the in-situ instruction had the fewer errors than the video instruction in all levels of complexity (see Fig. 6a). The difference was not significant in the 8-step ($t(15) = 1.86$, $p > 0.05$, $r = 0.43$) and 16-step constructions ($t(16.84) = 1.94$, $p > 0.05$, $r = 0.42$). But, the difference was statistically significant in the 24-step construction ($t(15) = 2.48$, $p < 0.05$, $r = 0.53$) and the 32-step construction ($t(15) = 2.31$, $p < 0.05$, $r = 0.50$). The effect size estimate indicates that the effect on ER for all four models using the provided instructions is large.

The comparison of TCTs revealed that the difference between both approaches was significant for all steps except for the 8-step construction ($t(30) = 1.11$, $p > 0.05$,

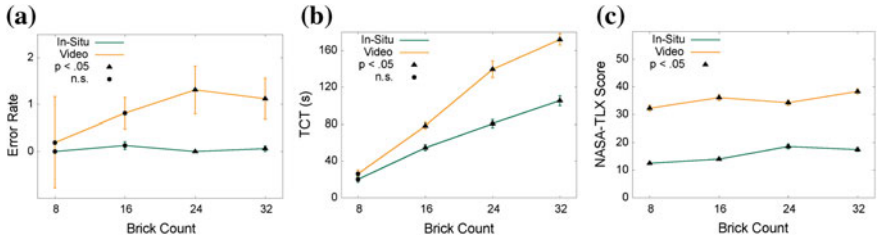

Fig. 6 The results of the lab study for constructions with different number of steps: **a** error rate (ER), **b** task completion time (TCT), and **c** NASA-Task Load Index (NASA-TLX) score

$r = 0.20$). Figure 6b shows the average TCT for the four constructions using the two instruction methods. In all cases the TCT was significantly faster using the in-situ approach (for the 16-step montage: $t(30) = 4.69$, $p < 0.001$, $r = 0.65$; for 24-step montage: $t(30) = 5.67$, $p < 0.001$, $r = 0.71$; for 32-step montage: $t(30) = 7.92$, $p < 0.001$, $r = 0.68$). The effect sizes show that effect of the provided instructions on the TCT of the assembly tasks is substantial except for the 8-step assembly task.

Further, the NASA-TLX scores statistically significantly differ in all four constructions (see Fig. 6c). In all cases the score for the in-situ instruction was significantly lower than the video approach: for the 8-step montage, $t(19.37) = 4.30$, $p < 0.001$, $r = 0.70$; for 16-step montage, $t(20.52) = 4.58$, $p < 0.001$, $r = 0.71$; for 24-step montage, $t(30) = 2.90$, $p < 0.007$, $r = 0.47$; for 32-step montage: $t(30) = 4.37$, $p < 0.001$, $r = 0.62$. The effect size estimate indicates that the effect on the perceived cognitive load using the two instruction approaches is large, and therefore substantial for all models.

4.3 Discussion

The results of the analysis reveal that there are significant differences between the video instruction and the in-situ projection approach in the ER, the TCT, and the perceived cognitive load during the assembly tasks. Using the in-situ system, the ER decreases up to 17%, the TCT is up to 1.5 times faster, and the perceived cognitive load is reduced up to 60% in comparison to the video-based instruction.

Further, the comparison of the in-situ and video-based instructions in different levels of complexity unveil that the in-situ instruction outperforms the video-based approach independent of the number of steps. In all levels the ER is lower and the TCT is faster. These differences are significant when the number of steps in the assembly task increase. On the other hand, the perceived cognitive load is significantly lower for the in-situ instruction independent from the number of steps in the assembly task.

5 Study #2: Creating Assembly Instructions

To evaluate our system for creating assembly instructions in a real world scenario, we conducted a user study using a real assembly task (a refurbished car's engine starter) with industrial workers. We made this conscious choice to increase the validity of the results, even if it is harder to reproduce the results. Using students and a lab-based study is in our view not appropriate in to address this questions.

5.1 Method

We used a repeated measures design with three conditions for creating an instruction: by demonstration using our system, using the editor, and video recording. The only independent variable was the creating-method. As dependent variables, we measured the task completion time (TCT) for creating instructions and the NASA-TLX score [25]. The order of the conditions was counterbalanced.

For the editor condition, we re-implemented the system presented by Korn et al. [29]. In contrast to our system, the user should use a graphical user interface (GUI) to manually highlight the bins, the workpiece carrier, or tools that have to be used for the assembly task using different geometric shapes (see Fig. 7a). Further, the GUI is used to define actions in each step of the assembly and create an instruction. For the video condition, we recorded a video of the assembly from the worker's point of view. A camcorder was installed behind the user in such a way that the worker's point of view could be simulated. The participant had to inform the experimenter when the video recording should be started and stopped.

As the assembly task, we chose the assembly of a car's engine starter (see Fig. 1). The task consisted of five steps and in each step one part should be assembled. When all five parts were put together on the workpiece carrier, the worker should fix two screws on top of the starter using a screwdriver.

We carried out the study in a car manufacturing company in Germany. After welcoming the participant and explaining the course of the study, we collected the demographics. Next, we introduced the participant to the workpiece carrier and let them get familiar with it. We allowed the participant to assemble the engine starter twice to get themselves familiar before starting the study. Afterwards, the study was started and participants had to create instructions using the three approaches. End of each condition, the experimenter measured the TCT. Afterwards, the participant completed the NASA-TLX questionnaire. At the end, we collected qualitative feedback through semi-structured interviews.

We recruited 10 workers from the company (2 female, 8 male), who were familiar with the engine starter. The participants were aged between 17 and 53 years (M = 32.1, SD = 13.9). All participants had experience in assembling the engine starter for at least one year and could be considered as experts.

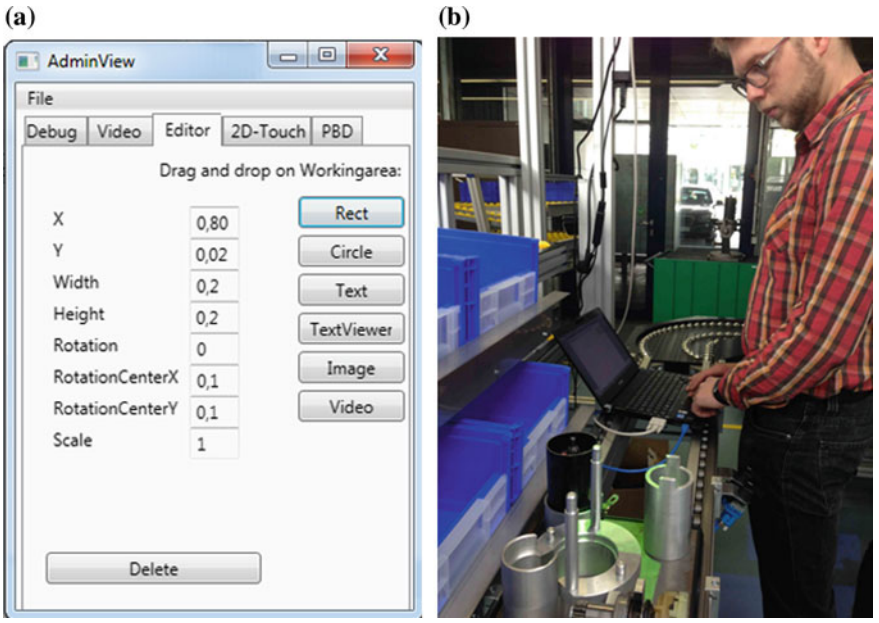

Fig. 7 **a** The graphical editor allows changing the properties of projected elements. **b** The worker can adjust the projection directly on the workpiece carrier

5.2 Results

We statistically compared the TCT between the methods. Mauchly's test indicated that the assumption of sphericity had been violated ($\chi^2(2) = 18.04$, $p < 0.0001$). Therefore, the degree of freedom was corrected using Greenhouse-Geisser estimates of sphericity ($\epsilon = 0.52$). A repeated measures ANOVA showed that there is a statistically significant difference in TCT between the methods ($F(1.05, 9.49) = 256.04$, $p < 0.0001$, $r = 0.97$). The effect size estimate reveals a large and therefore substantial effect. Post hoc tests using Bonforroni correction revealed a significant difference between all three methods (all $p < 0.05$). The video method had the shortest TCT ($M = 0.58$ min, $SD = 0.08$) followed by the PbD ($M = 1.52$ min, $SD = 0.63$) and the editor ($M = 16.16$ min, $SD = 3.07$). The results are also depicted in Fig. 8a.

Further, we statistically compared the NASA-TLX scores between the methods (see Fig. 8b). The sphericity assumption was not violated ($p > 0.05$). A repeated measures ANOVA determined that the methods used had a statistically significant effect on the NASA-TLX score ($F(2, 18) = 19.83$, $p < 0.0001$, $r = 0.81$). The effect size estimate shows a large and substantial effect. Post hoc tests using the Bonferroni correction revealed that the editor had a statistically significantly higher perceived cognitive load ($M = 23.10$, $SD = 7.79$, $p < 0.007$) than PbD ($M = 11.80$, $SD = 5.22$) and the video ($M = 10.40$, $SD = 7.07$, $p < 0.001$). However, the difference between PbD and video was not statistically significant ($p > 0.05$).

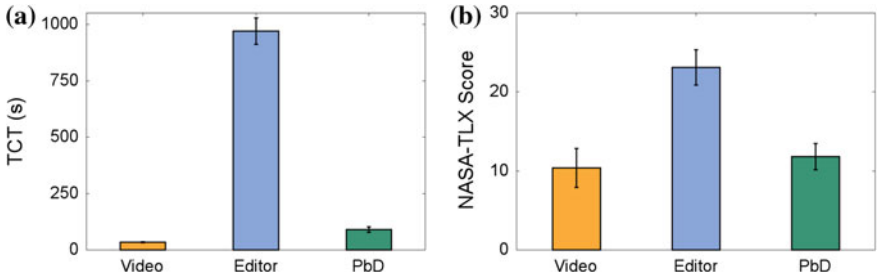

Fig. 8 The results of the user study for creating assembly instructions **a** The task completion time across the different approaches. **b** The perceived cognitive load across the approaches using the NASA-TLX. The *error bars* depict the standard error

The qualitative feedback showed that the participants found the editor hard to use. Although they were experts in assembling an engine starter, they didn't have enough experience in using a computer (e.g., P6, P1). Further, a participant stated "*using the editor is too time-consuming*" (P3). One participant had also privacy concerns when recording a video as co-workers could identify him based on his hands and his wristwatch (P4).

5.3 Discussion

The results of the study reveal that the editor approach requires significantly higher perceived cognitive load compared to the PbD and video approaches. Whereas, there is no significant difference in perceived cognitive load required for creating assembly instructions using the PbD and video approaches. Hence, the additional perceived cognitive load added due to the use of our interactive system is not significant.

On the other hand, the results show that recording the video is the fastest way for creating an assembly instruction followed by PbD and the editor approach. One reason is that no additional time is required to capture the depth information after each assembly step. In contrast, the PbD approach requires that the users shortly remove their arms and head from the work area to capture the depth data of the product.

Although the PbD-based and video-based approaches are faster and require less cognitive effort than the editor-based approach in creating instructions, the approaches might differ when assembling the engine starter. Therefore, we conducted a followup study to evaluate the instructions while assembling the engine starter with novice users.

6 Study #3: Evaluation of Assembly Instructions

In the previous study we assessed different approaches for creating assembly instructions. In order to evaluate the practicality of the approaches in assembling a product, we conducted a followup study assembling the same engine starter we used in the previous study using the previously created instructions.

6.1 Method

For providing assembly instructions we used the instructions created in the previous study. We randomly chose one instruction created using each approach resulting in three instructions: (1) the video-based assembly instruction, (2) the in-situ projection instruction created using the editor, (3) the in-situ projection instruction created using PbD. For the in-situ projection instruction using the editor, the user explicitly created the instruction using a graphical editor. In contrast, our system automatically generated the other instruction. We chose a between subject design with three groups to prevent a learning effect between the different instructions. The only independent variable that differed between the groups was the type of instruction. As dependent variables we measured the number of errors (ER), the task completion time (TCT), and the NASA-TLX score.

We conducted the study in the same company as in the previous study. After welcoming the participant and explaining the course of the study, we collected the demographics and ensured that the participant never assembled an engine starter before. Then, the participant was accompanied to our prototype and one of the instructions was assigned and explained. As the participants did not differ in skills, the condition was randomly assigned. The participant was told to assemble an engine starter based on the instructions provided. When the participant was ready, the experimenter started the instruction and counted the ER. The TCT was measured by the system automatically. During the assembly the experimenter did not provide any help. After the assembly was done, the participant was asked to fill in a NASA-TLX questionnaire. Finally, qualitative feedback was collected through a semi-structured interview.

We recruited 51 participants (12 female, 39 male) aged between 23 and 60 years ($M = 47.8$, $SD = 9.3$). We divided the participants equally between the conditions, resulting in 17 participants per condition. All participants were employees of the company and were unfamiliar with the assembly task and the product, i.e., assembling an engine starter. Hence, they can be considered novice users. None of the recruited participants took part in the previous study.

6.2 Results and Discussion

We statistically compared the ER, the TCT and NASA-TLX between the groups. The assumption of homogeneity of variance had not been violated ($p > 0.05$). A one-way ANOVA test revealed no significant effect on ER between the groups ($F(2, 48) = 0.89, p > 0.05$). The group using the instruction created by our system had the lowest ER ($M = 1.12$, $SD = 0.86$) followed by the group using the instruction created by the editor ($M = 1.24$, $SD = 0.90$) and the group using the video-based instruction ($M = 1.53$, $SD = 1.01$). Results are also depicted in Fig. 9b

The statistical analysis also revealed no significant difference in the TCT between the groups ($F(2, 48) = 0.32$, $p > 0.05$). According to Fig. 9a, the group using the instruction created by our system had the shortest TCT ($M = 2.21$ min, $SD = 1.05$) and the group using the instruction created with the editor had the longest TCT ($M = 2.52$ min, $SD = 1.39$). The group using the video instruction took on average 2.22 min ($SD = 1.31$) to assemble the product.

The analysis showed no significant effect on the NASA-TLX score between the groups ($F(2, 48) = 1.38, p > 0.05$). The group using the video-based instruction had the lowest perceived cognitive load ($M = 20.59$, $SD = 13.90$) followed by the group using the instruction created using our system ($M = 27.53$, $SD = 13.87$) and using the editor ($M = 28$, $SD = 15.84$). A graphical representation is depicted in Fig. 9c.

The qualitative feedback indicated that the projected instructions were generally well perceived. They particularly found the step by step feedback of the projected instructions very helpful (P42, P33). Additionally they mentioned that directly projected feedback onto the workplace was very useful (P30, P12). One participant stated that "I don't have to think anymore while working" (P24). Another participant mentioned that "I would rather work autonomous in the daily life, but for training I would use it" (P45). Participants using the video instruction mentioned that the video was helpful for learning the task instead of having an instructor (P22, P51) but they didn't want a video playing all day (P37, P25).

The analysis shows that the ER is reduced and the TCT is faster in the assembly task using our system compared to the other two approaches. However, the change is not significant. The results indicate that the instruction automatically created using

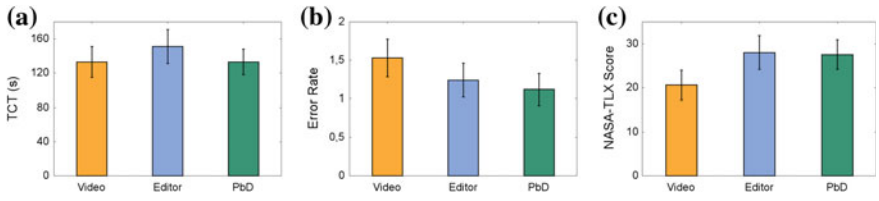

Fig. 9 The results of the user study for evaluating the previously created assembly instructions **a** the task completion time across the different approaches. **b** The error rate for the different approaches. **c** The perceived cognitive load across the approaches using the NASA-TLX. The *error bars* depict the standard error

our system slightly performs better than the explicitly created instruction using the editor. On the other hand, the results suggest that the in-situ projection increases the perceived cognitive load required during the assembly, but the difference is not significant. The qualitative feedback indicates that the step by step instructions provided directly in the work area through the in-situ projection is more accepted than a video-based instruction.

As the assembly instruction only consisted of five steps, no big differences were expected. Based on the results from study 1 it can be expected that with more steps the differences between these instructions would increase and a clearer advantage for the in-situ instructions would be expected to show.

7 Implications

The aforementioned user studies revealed implications on both creating assembly instructions and performing an assembly task based on previously created instructions. In the following we discuss the insights gained through these studies.

7.1 Creating Assembly Instructions

The results of the study indicate that creating instructions using the editor approach is more time-consuming and demands more cognitive load compared to the PbD approach and the video-based approach.

Using the PbD approach, the time required to create an instruction is higher than the video recording approach as our system requires the user to wait 1.5 s between each step for detecting that a step was performed. The time is even higher when using the editor approach as the user has to manually specify each step. However, editing steps in both PbD approach and editor approach is easier than editing video-based instructions since each step can be modified separately. In contrast, video-based instructions need to be post-processed and manually edited. Editing videos can be complex and may result in re-recording the video even if only a single step needs to be altered. Another advantage of both in-situ approaches is that it records the depth information of each step. Using the depth information the system can monitor if the correct part is picked and if it is assembled correctly.

While the video approach has the lowest perceived cognitive load when creating instructions, the results further show that using the PbD approach does not significantly increase the mental load. However, the editor approach induces a higher perceived cognitive load by interacting with GUIs.

A further advantage of including semantic information into the instructions is that the instructions can be targeted to a specific work place automatically. One could even imagine a skilled worker in one company (or one country) can create

the instructions and these instructions can then be downloaded to an assembly table in another company (or country) (cf. [17]).

7.2 Assembly Performance

When it comes to assembling a product, the results suggest that the in-situ projection approach reduces the mental load of the worker, the TCT, and the ER. Specially, these effects are significant when the number of assembly steps increase. As the in-situ assembly instructions are provided directly in the work area, the distraction is minimized compared to showing the videos on a monitor close to the work area. This reduces the cognitive effort that is required for following instructions and also reduces the TCT for assembling a product. Furthermore, our system's step by step error control can monitor if the correct part is picked and if it is assembled correctly using depth information. This leads to fewer errors even when the number of steps increases.

7.3 Limitations

It should be mentioned that the proposed PbD system has certain limitations. The current version of the system tracks only one assembly part per work step. This process is favored by the industry as it is less error prone than assembling multiple parts in a single step. However, the system can be easily extended to track more than just a single item per step. Furthermore, all assembled parts on the workpiece carrier should be visible to the top-mounted Kinect to monitor the assembly task. Therefore, the workpiece carrier has to be designed to support this setup.

8 Conclusion

In this chapter, we presented a system that leverages the concept of PbD to create semantically-rich assembly instruction for enabling assistive augmentation at the workplace. The proposed system enables process engineers who are creating assembly instructions to create instructions faster than using a graphical editor for creating assembly instructions. In contrast to just recording video, which is slightly faster, our proposed system retains all features of interactive instructions and does not add any significant perceived cognitive load to the worker using the instructions for learning assembly steps in comparison to watching video instructions. The system was evaluated with experts in a production environment using a real product.

The system provides instructions using in-situ projection directly in the work area. It highlights a bin where a part should be picked from and shows the position where

the part should be assembled. In a large laboratory study, we could show that in-situ instructions outperform the video-based instruction in assembly tasks with different numbers of steps. It decreases the error rate, the task completion time, and the perceived cognitive load. This was also validated in a real assembly environment.

Creating such interactive instructions based on demonstration is not only limited to the assembly work place. It could be easily ported to other application domains. We currently explore further domains, in particular the home environment, where we assess if such a system could teach persons with learning disabilities to learn basic skills for independent living, such as cooking [14, 32] and cleaning their home.

Acknowledgements This work is funded by the German Federal Ministry for Economic Affairs and Energy in the project motionEAP [20], grant no. 01MT12021E. We thank Mathias Hoppe for his work in helping to implement the software and conducting the user study. We further thank Klaus Klein, Michael Spreng, and Johann Hegel from Audi AG.

References

1. Aleotti J, Caselli S (2006) Robust trajectory learning and approximation for robot programming by demonstration. Robot Auton Syst 54(5):409–413
2. Antifakos S, Michahelles F, Schiele B (2002) Proactive instructions for furniture assembly. In: UbiComp 2002: ubiquitous computing. Springer, pp 351–360
3. Bannat A, Gast J, Rigoll G, Wallhoff F (2008) Event analysis and interpretation of human activity for augmented reality-based assistant systems. In: 4th international conference on intelligent computer communication and processing, 2008. ICCP 2008. IEEE, pp 1–8
4. Barna J, NovakovaMarcincinova L, Novak-Marcincin J, Fecova V, Janak M, Torok J (2012) Open source tools in assembling process enriched with elements of augmented reality. In: Proceedings of the 2012 virtual reality international conference. ACM, p 2
5. Bay H, Tuytelaars T, Van Gool L (2006) Surf: speeded up robust features. In: Computer vision-ECCV 2006. Springer, pp 404–417
6. Billard A, Calinon S, Dillmann R, Schaal S (2008) Robot programming by demonstration. In: Springer handbook of robotics. Springer, pp 1371–1394
7. Blanke U, Schiele B, Kreil M, Lukowicz P, Sick B, Gruber T (2010) All for one or one for all? Combining heterogeneous features for activity spotting. In: 2010 8th IEEE international conference on Pervasive computing and communications workshops (PERCOM Workshops). IEEE, pp 18–24
8. Büttner S, Sand O, Röcker C (2015) Extending the design space in industrial manufacturing through mobile projection. In: Proceedings of the 17th international conference on human-computer interaction with mobile devices and services adjunct. ACM, pp 1130–1133
9. Büttner S, Mucha H, Funk M, Kosch T, Aehnelt M, Robert S, Röcker C (2017) The design space of augmented and virtual reality applications for assistive environments in manufacturing: a visual approach. In: Proceedings of the 10th international conference on pervasive technologies related to assistive environments. ACM, pp 433–440
10. Büttner S, Funk M, Sand P, Röcker C (2016) Using head-mounted displays and in-situ projection for assistive systems: a comparison. In: Proceedings of the 9th ACM international conference on pervasive technologies related to assistive environments. ACM, p 44
11. Caudell TP, Mizell DW (1992) Augmented reality: an application of heads-up display technology to manual manufacturing processes. In: Proceedings of the twenty-fifth hawaii international conference on system sciences, vol 2. IEEE, pp 659–669

12. Collett T, MacDonald BA (2006) Developer oriented visualisation of a robot program. In: Proceedings of the 1st ACM SIGCHI/SIGART conference on human-robot interaction. ACM, pp 49–56
13. Fiorentino M, de Amicis R, Monno G, Stork A (2002) Spacedesign: a mixed reality workspace for aesthetic industrial design. In: Proceedings of the 1st international symposium on mixed and augmented reality. IEEE Computer Society, p 86
14. Funk M, Korn O, Schmidt A (2015) Enabling end users to program for smart environments. In: Proceedings of the CHI 2015—workshop on end user development in the internet of things era 12.2, pp 9–14
15. Funk M, Kosch T, Schmidt A (2016) Interactive worker assistance: comparing the effects of in-situ projection, head-mounted displays, tablet, and paper instructions. In: Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing. ACM, pp 934–939
16. Funk M, Mayer S, Schmidt A (2015) Using in-situ projection to support cognitively impaired workers at the workplace. In: Proceedings of the 17th international ACM SIGACCESS conference on computers and accessibility
17. Funk M, Schmidt A (2015) Cognitive assistance in the workplace. Pervasive Comput IEEE 14(3):53–55
18. Funk M, Kosch T, Greenwald SW, Schmidt A (2015) A benchmark for interactive augmented reality instructions for assembly tasks. In: Proceedings of the 14th international conference on mobile and ubiquitous multimedia. ACM, pp 253–257
19. Funk M, Bächler A, Bächler L, Korn O, Krieger C, Heidenreich T, Schmidt A (2015) Comparing projected in-situ feedback at the manual assembly workplace with impaired workers. In: Proceedings of the 8th ACM international conference on pervasive technologies related to assistive environments. ACM, p 1
20. Funk M, Kosch T, Kettner R, Korn O, Schmidt A (2016) Motioneap: an overview of 4 years of combining industrial assembly with augmented reality for industry 4.0. In: Proceedings of the 16th international conference on knowledge technologies and datadriven business
21. Funk M, Shirazi AS, Mayer S, Lischke L, Schmidt A (2015) Pick from here!: an interactive mobile cart using in-situ projection for order picking. In: Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing. ACM, pp 601–609
22. Guo A, Raghu S, Xie X, Ismail S, Luo X, Simoneau J, Gilliland S, Baumann H, Southern C, Starner T (2014) A comparison of order picking assisted by head-up display (HUD), cart-mounted display (CMD), light, and paper pick list. In: Proceedings of the 2014 ACM international symposium on wearable computers. ACM, pp 71–78
23. Hahn J, Ludwig B, Wolff C (2015) Augmented reality-based training of the PCB assembly process. In: Proceedings of the 14th international conference on mobile and ubiquitous multimedia. ACM, pp 395–399
24. Hardy J, Alexander J (2012) Toolkit support for interactive projected displays. In: Proceedings of the 11th international conference on mobile and ubiquitous multimedia. ACM, p 42
25. Hart SG, Staveland LE (1988) Development of NASA-TLX (task load index): results of empirical and theoretical research. Adv Psychol 52:139–183
26. Henderson S, Feiner S (2011) Exploring the benefits of augmented reality documentation for maintenance and repair. IEEE Trans Visual Comput Graph 17(10):1355–1368
27. Hermann M, Pentek T, Otto B. Design principles for industrie 4.0 scenarios: a literature review
28. Klompmaker F, Nebe K, Fast A (2012) dSensingNI: a framework for advanced tangible interaction using a depth camera. In: Proceedings of the sixth international conference on tangible, embedded and embodied interaction. ACM, pp 217–224
29. Korn O, Schmidt A, Hörz T (2013) Augmented manufacturing: a study with impaired persons on assistive systems using in-situ projection. In: Proceedings of the 6th international conference on pervasive technologies related to assistive environments. ACM, p 21
30. Korn O, Funk M, Abele S, Hörz T, Schmidt A (2014) Context-aware assistive systems at the workplace: analyzing the effects of projection and gamification. In: Proceedings of the 7th international conference on pervasive technologies related to assistive environments. ACM, p 8

31. Kubitza T, Schmidt A (2015) Towards a toolkit for the rapid creation of smart environments. In: End-user development. Springer, pp 230–235
32. Lee C-H, Bonnani L, Selker T (2005) Augmented reality kitchen: enhancing human sensibility in domestic life. In: ACM SIGGRAPH 2005 posters. ACM, p 60
33. Lieberman H (2001) Your wish is my command: programming by example. Morgan Kaufmann
34. Linder N, Maes P (2010) LuminAR: portable robotic augmented reality interface design and prototype. In: Adjunct proceedings of the 23nd annual ACM symposium on user interface software and technology. ACM, pp 395–396
35. Lucke D, Constantinescu C, Westkämper E (2008) Smart factory-a step towards the next generation of manufacturing. In: Manufacturing systems and technologies for the new frontier. Springer, pp 115–118
36. Marinos D, Wöldecke B, Geiger C (2013) Prototyping natural interactions in virtual studio environments by demonstration: combining spatial mapping with gesture following. In: Proceedings of the virtual reality international conference: laval virtual. ACM, p 2
37. Myers BA (1986) Creating dynamic interaction techniques by demonstration. In: ACM SIGCHI bulletin 17.SI, pp 271–278 (1986)
38. Pinhanez C (2001) The everywhere displays projector: a device to create ubiquitous graphical interfaces. In: Ubicomp 2001: ubiquitous computing. Springer, pp 315–331
39. Rüther S, Hermann T, Mracek M, Kopp S, Steil J (2013) An assistance system for guiding workers in central sterilization supply departments. In: Proceedings of the 6th international conference on pervasive technologies related to assistive environments. ACM, p 3
40. Salonen T, Sääski J, Hakkarainen M, Kannetis T, Perakakis M, Siltanen S, Potamianos A, Korkalo O, Woodward C (2007) Demonstration of assembly work using augmented reality. In: Proceedings of the 6th ACM international conference on image and video retrieval. ACM, pp 120–123
41. Schmidt A (2000) Implicit human computer interaction through context. Pers Technol 4(2–3):191–199
42. Tang A, Owen C, Biocca F, Mou W (2003) Comparative effectiveness of augmented reality in object assembly. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 73–80
43. Ward JA, Lukowicz P, Troster G, Starner TE (2006) Activity recognition of assembly tasks using body-worn microphones and accelerometers. IEEE Trans Pattern Anal Mach Intell 28(10):1553–1567
44. Wilson AD (2004) TouchLight: an imaging touch screen and display for gesture-based interaction. In: Proceedings of the 6th international conference on Multimodal interfaces. ACM, pp 69–76
45. Wilson AD (2010) Using a depth camera as a touch sensor. In: ACM international conference on interactive tabletops and surfaces. ACM, pp 69–72
46. Wohlgemuth W, Triebfürst G (2000) ARVIKA: augmented reality for development, production and service. In: Proceedings of DARE 2000 on designing augmented reality environments. ACM, pp 151–152
47. Zauner J, Haller M, Brandl A, Hartman W (2003) Authoring of a mixed reality assembly instructor for hierarchical structures. In: The second IEEE and ACM international symposium on mixed and augmented reality, 2003. Proceedings. IEEE, pp 237–246
48. Zollner R, Rogalla O, Dillmann R, Zollner M (2002) Understanding users intention: programming fine manipulation tasks by demonstration. In: IEEE/RSJ international conference on intelligent robots and systems, 2002, vol 2. IEEE, pp 1114–1119