Robot Performing Peg-in-Hole Operations by Learning from Human Demonstration

Zuyuan Zhu*, Huosheng Hu†, Dongbing Gu‡
School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK
*z.zhu@essex.ac.uk, †hhu@essex.ac.uk, ‡dgu@essex.ac.uk

***Abstract*—This paper presents a novel approach for a robot to conduct assembly tasks, namely robot learning from human demonstrations. The learning of robotic assembly task is divided into two phases: teaching and reproduction. During the teaching phase, a wrist camera is used to scan the object on the workbench and extract its SIFT feature. The human demonstrator teaches the robot to grasp the object from the effective position and orientation. During the reproduction phase, the robot uses the learned knowledge to reproduce the grasping manipulation autonomously. The robustness of the robotic assembly system is evaluated through a series of grasping trials. The dual-arm Baxter robot is used to perform the Peg-in-Hole task by using the proposed approach. Experimental results show that the robot is able to accomplish assembly task by learning from human demonstration without traditional dedicated programming.**

***Index Terms*—learning from demonstration, robotic assembly, machine learning, Peg-in-Hole task, Baxter robot**

I. Introduction

Robotic assembly needs a high degree of repeatability, flexibility, and reliability to improve the automation performance in assembly lines. Traditionally, the robotic assembly operation is programmed or hard-coded by human operators with a good knowledge of all geometrical characteristics of individual parts. This assembly operation is normally position-controlled and designed to follow desired trajectories with an extremely tight positional accuracy [1]. Similar to humans performing compliant movements by using force feedback and tactile information, the contacts and forces are sensed by robot sensors and used to implement the assembly procedures. In general, the current robotic assembly systems can handle known objects within the well-structured assembly lines very well.

For instance, a multi-robot coordinated assembly system for furniture assembly was investigated by Knepper et al. in [2]. They listed the geometry of individual parts in a table so that a group of robots can conduct parts delivery or parts assembly collaboratively. The furniture parts were predefined in CAD files for the modeling and recognition purpose so that the correct assembly sequence can be deduced from geometric data. On the other hand, Suarez-Ruiz and Pham proposed a taxonomy of the manipulation primitives for bi-manual pin insertion, which was only one of the key steps in the autonomous assembly of an IKEA chair [3].

However, when the assembly tasks change, the current robotic assembly need tedious reprogramming for every new workpiece before the operation. In contrast, Learning from Demonstration (LfD) paradigm enables robots to learn the involved forces and trajectories for assembling tasks from human demonstrations. The LfD allows for creating a connection between perception and action for the robot. Recently, LfD has been suggested as an effective way to accelerate the programming of learning processes from the low-level control to the high-level assembly planning [4]. Therefore, LfD is a preferable approach for robotic assembly tasks [5].

In this paper, we propose a new approach to solve one of the assembly tasks, Peg-in-Hole (PiH) problem, by using the LfD paradigm. The object to be assembled is not limited to predefined objects. The geometrical characteristics of the parts are not necessary prior knowledge. In addition, the objects can be placed in arbitrary poses and positions within the workspace of the robotic arm. The robot learns assembly skills through LfD paradigm, which allows non-experts to teach the robot how to assemble. Instead of imitating the trajectories demonstrated by the human, the robot learns the most important position information of the PiH task through the kinesthetic teaching.

The rest of this paper is organized as follows. Section II briefly presents the related work on the field of robotic assembly and explain how the assembly problem has been solved up to now. In Section III, we present the methods that we used to solve the assembly problem from two aspects: (i) how the demonstrator teaches the robot and (ii) how the robot reproduces the learned skills. Then, experimental evaluation of the object recognition and assembly of Lego blocks are given in Section IV to demonstrate the feasibility and performance of the proposed approach. Finally, a brief conclusion and the future improvement are given in Section V.

II. Related Work

PiH is one of the most essential and representative assembly tasks and has been widely researched [6]–[8]. It is a process that a robotic gripper grabs the peg and inserts it in a hole. The positioning inaccuracies and tight tolerances between the peg and the hole involved in PiH operations require some degree of online adaptation of the programmed trajectories. Up to today, a number of robotic assembly systems were proposed to solve the PiH problem, and most of them use additional specialized force sensors, markers and/or cameras.

Nemec et . al [9] proposed an approach to acquire not only trajectories but also forces and torques occurring during the task demonstration. During the human demonstration phase, the Cartesian space trajectory and the associated force/torque profile of the human motion of the PiH task are recorded. In the reproduction phase, the robot uses admittance or impedance control law to adapt to the desired forces and reduce the force/torque error.

Tang et. al [10] introduced Gaussian Mixture Regression (GMR) to learn the state-varying admittance directly from human demonstration data. The demonstration data is collected by a specially designed device, where force/torque sensor is embedded to collect the wrench information and active makers are placed to record the corrective velocity that human applies on the peg.

Instead of learning from human demonstrators, Kramberger et. al [11] proposed an algorithm to learn geometrical constraints between the parts and their final locations from the experiments executed by a real robot. The robot tries to insert the available pegs into different holes, if the action is executed successfully, then the robot learns that the peg fits in the hole. The judgment of success or failure is accomplished by using force/torque data and poses extracted by vision.

In addition, the peg would usually occlude the hole when the robot approaches the hole during the peg-in-hole operation. Therefore, vision-based pose estimation is not suitable for the high-accuracy assembly tasks in which two parts occlude each other. If the camera is mounted on the robotic arm, the occlude problem can be eliminated, but additional sensory data is needed to estimate the camera pose [12].

To correct the pose of assembly parts, Xiao et al. devised a nominal assembly-motion sequence to collect data from exploratory complaint movements [13]. The data are then used to update the subsequent assembly sequence to correct errors in the nominal assembly operation. Nevertheless, the uncertainty in the pose of the manipulated object should be further addressed in the future research.

III. Methods

This section is structured as follows. We begin with the analysis of object mapping, first explaining how the object is mapped and how the robot learns effective grasping pose. We continue by showing how the human demonstrator teaches the robot to learn the assembly skill.

A. Teaching

The object detection runs on a stock Baxter with a development workstation. We map one side of an object with Baxter's wrist camera, then use the learned model to detect the object, localize it and pick it up [14]. The object detection and pose estimation uses conventional computer vision algorithms like SIFT and kNN for feature abstraction and object classification. The object classes are based on the specific object rather than general categories. Each object has a unique name labeled by the human.

The learning process is presented in Figure 1. Firstly, the wrist camera captures images of the object and extract bounding boxes for the objects. Then extract the features of the bounding boxes which are used to represent an object class. Last, the human demonstrator teaches the robot how to grasp the object from an effective pose by kinesthetic guiding.

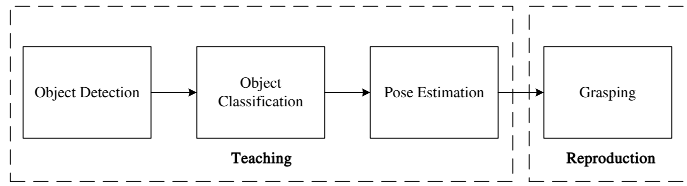


Fig. 1: The mapping of the object.

1) Object Detection: We use the grey background to reduce the reflective light which introducing noise to the detection of the object. During detection, the wrist camera moves along a line over the object at a fixed height. For an input image, the robot extracts bounding boxes for the object in the workspace. The smallest bounding box which contains the object is selected. The extracted object is shown in Figure 2.

2) Object Classification: In the object classification module, the bounding boxes are further abstracted with SIFT features. The k-means algorithm is used to extract a visual vocabulary of the SIFT feature. Then a Bag of Words feature is constructed for each image. Next, the Bag is augmented with a histogram of colors included in the image. The augmented feature vector is learned by the robot and labeled by the human with an intuitive name, like RedLegoBlock.

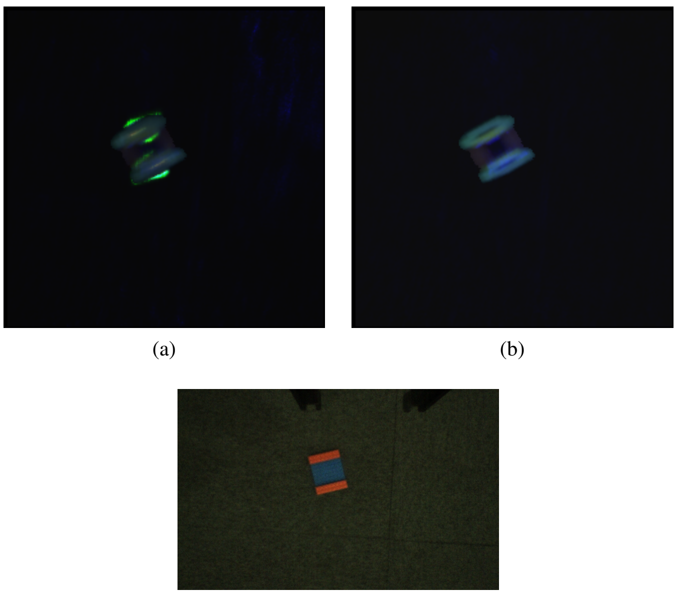
3) Pose Estimation: The robot has learned to detect and localize the object in Section III-A1 and III-A2. Based on the learned information, the robot could reach and grasp the object. However, the grasping pose is optimized and not efficient enough. We improve the grasping efficiency by teaching the robot effective grasping poses, i.e., the human demonstrator guide the robot's arm to the grasping pose (see Figure 5a). As an object can be grasped by more than one pose, the human demonstrator teaches the robot more than one pose to ensure the robot has more options if it fails the first time.

B. Reproduction

In the reproduction phase, the robot uses the learned knowledge to reproduce the grasping autonomously, which consists of three phases as follows:

- First, the robot uses the wrist camera to scan and detects the object in the workspace. From the input images, the robot extracts bounding boxes of the object. Then, the robot uses the bounding boxes to extract the augmented feature vector as described in Section III-A2. Next, the vector is incorporated into a k-nearest-neighbors model which is used to classify objects and output the label. This label is used to identify the object and refer to other information about the object for grasping.

- Next, to estimate the pose, the robot requires a crop of the image gradient of the object at a specific and known pose. The robot rotates the training image and find the closest match to the image currently learned in Section III-A1 and III-A2.
- Last, once the grasping pose is determined, the robot need to identify the grasping point. The grasping module is a linear model that estimates the grasping success. The module takes the 3D pose of the object as input and outputs the grasping point (x, y, θ). The (x, y) is the 2D position in the plane of the table. The accurate height of the gripper does not matter, as the gripper always start from 38 cm over the table and approaches the table gradually until hitting the table, triggering the grasping. The θ is the angle which the gripper assumes for grasping.



(a) (b) (c)

Fig. 2: Object detection in different views: (a) the discrepancy view shows differences between the observed scene and the background, i.e., the object; (b) the standard deviation view of the object shows the edges of the object; (c) the object in the wrist camera view.

C. Pseudo Code

In this section, we give a brief outline of the robot program to implement the proposed approach described above. It is presented here in the format of pseudocode, see **Algorithm** 1.

In the grasping part of the pseudocode, O_i is the new object to be mapped; Z_c is the "components zone". During the teaching of grasping skill, it should be noted that the pose $P(x, y, \theta)$ is relative to the camera's orientation, which is recorded in θ. When the robot reproduces the grasping, the robot rotates the camera to find the closest match to the learned image and pose. $\mathbf{V}_{f,i}$ is the feature extracted in the III-A2 *Object Classification* step.

In the assembly part of the pseudocode, S_k is the sequence motion of assembly task demonstrated by the human.

In the reproduction part of the pseudocode, O'_i is the learned object; Z_a is the "assembly zone"; O'_m is the former detected object to be grasped; O'_n is the later detected object to be assembled; F_p is the pressing force that the arm applies on the two objects; F_0 is the threshold force, which controls the insertion movement.

Algorithm 1 Pseudocode of the robotic assembly using LfD

1: initialize
2: */* line 3 - 10: learning from demonstration: grasping */*
3: **for** O_i in Z_c; $i \in [1, N]$ **do**
4: detect the bounding box B_i;
5: extract feature vector $\mathbf{V}_{f,i}$ from B_i;
6: **for all** demonstration D_j; $j \in [1, M]$ **do**
7: human demonstrates the grasping pose $P(x, y, \theta)$;
8: robot maps pose $P(x, y, \theta)$ and feature $\mathbf{V}_{f,i}$;
9: **end for**
10: **end for**
11: */* line 12 - 13: learning from demonstration: assembly*/*
12: human demonstrates the assembly;
13: robot learns the motions and sequence $S_k, k \in [1, 3]$;
14: */* line 15 - 31: robot reproduces assembly task*/*
15: **for** O'_i in Z_c; $i \in [i, N]$ **do**
16: detect and classify the object O'_m;
17: grasp object O'_m;
18: break;
19: **end for**
20: assembly sequence S_1: move object O'_m to zone Z_a;
21: **for** O'_i in Z_c; $i \in [i, N]$ **do**
22: detect and classify the object O'_n;
23: grasp object O'_n;
24: break;
25: **end for**
26: assembly sequence S_2: move object O'_n to zone Z_a;
27: assembly sequence S_3: assemble object O'_n with object O'_m;
28: **while** $F_p < F_0$ **do**
29: F_p++;
30: **end while**
31: aseembly done;

IV. Experimental Results

In this section, the object recognition and LfD-based robotic assembly systems are both evaluated. Figure 3 shows three kinds of Lego blocks used for the evaluation of object recognition and grasping performance of the system: namely Yellow Lego block, Red Lego block, and RedBlue Lego block. The Red Lego block and the Yellow Lego block are same in dimension, i.e. $2\frac{1}{2} \times \frac{1}{2} \times \frac{7}{16}$ *inch*. The RedBlue Lego block is composed of a group of red and blue Lego blocks. The dimension is $2\frac{1}{2} \times 3\frac{1}{8} \times \frac{13}{16}$ *inch*.

Figure 4 shows the Baxter research robot used to conduct the experiments for all the research work in this paper. The

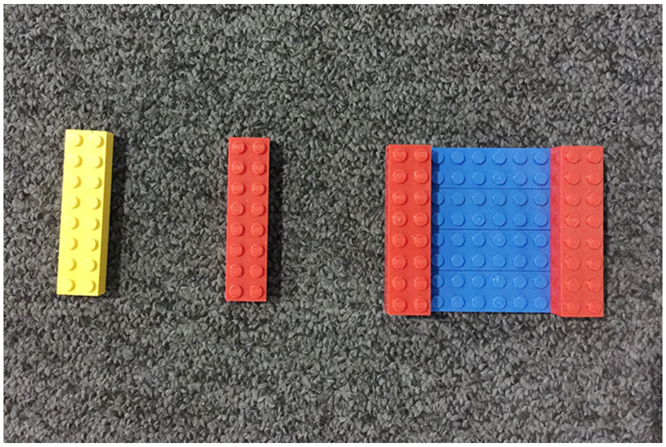


Fig. 3: Evaluation objects from left to right: Yellow Lego block, Red Lego block, and RedBlue Lego block.

camera built-in wrist of the Baxter robot can capture images at the maximum resolution of 1280×800. However, we only used an effective image resolution of 640×400 with the same field of view. Baxter's arms are also loaded with Infrared Range (IR) Sensors which has the maximum range of 0.4m and minimum range of 0.04m. The arm of Baxter has seven degree-of-freedom (DOF), but the arm always keeps crane pose to capture consistent views of the object and makes the picking problem simple.

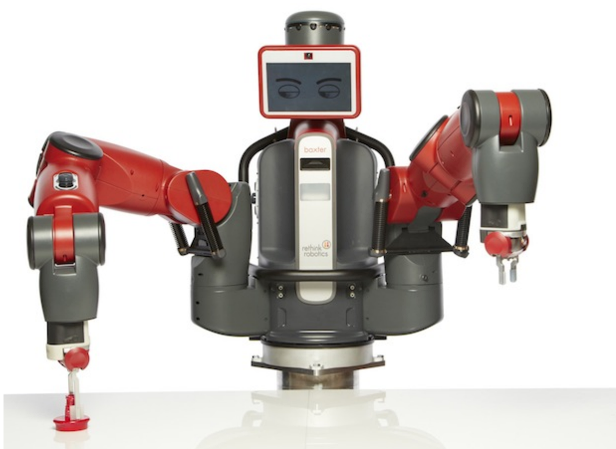


Fig. 4: The dual-arm Baxter Robot built by Rethink Robotics.

A. Object Recgonition and Picking Task

The object recognition and picking task assess the ability of the robot to learn efficient picking pose from human demonstrations. The robot arm was set at crane pose and kept this pose during the whole experiment. In the beginning, the robot arm located at the height of 38 cm above the table. Before recognition, the object to be recognized was placed under the robot's wrist camera for scanning. The features of the object were abstracted by Line Scan. The robot moved its arm 28 cm back and forth above the object to make a synthetic photograph during the line scan. Next, the object's position was estimated by using image matching in the synthetic photograph.

Then, the object was labeled by the human, like RedLegoBlock. The robot knew the position of the object and could plan a grasp trajectory using inverse kinematics solver. However, for some objects, the best grasp point is not the geometry centre. For example, the RedBlue Lego block (see Figure 3) can only be gripped from the edge as the object is too big for Baxter's gripper to grip around the object's centre. In this paper, we implemented the LfD in the learning of the picking task, see Figure 5. The human demonstrator teaches the robot to grip the RedBlue Lego block from the edge by kinesthetic guiding. The robot learns the successful picking pose.

For each trial, we placed the object at a random location on the table within approximately 25 cm of the wrist camera's view centre. In this paper, we evaluated the recognition ability with three different objects, see Figure 3. Each object was tested for 30 times, the result is listed in Table I. From the Table, we can see that the performance of the picking ability is generally good. The failure of the RedBlue Lego block is due to the gripper's motor noise during grasping.

TABLE I: Object Recognition and Picking Experiments

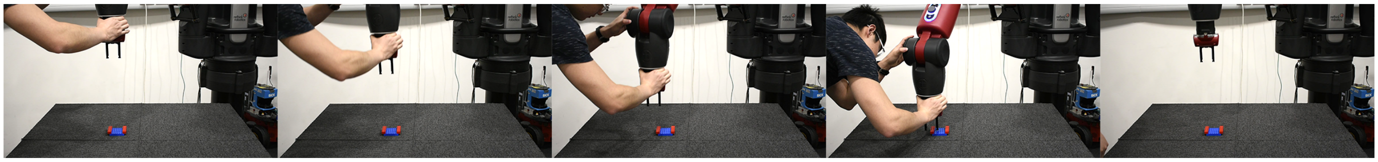
Picking Objects	**Picking Performance**		
	Picking Times	***Successful Times***	***Successful Rate***
Red Lego Block	30	29	96.7%
Yellow Lego Block	30	30	100%
RedBlue Lego Block	30	27	90%

B. Lego Blocks Assembly Task

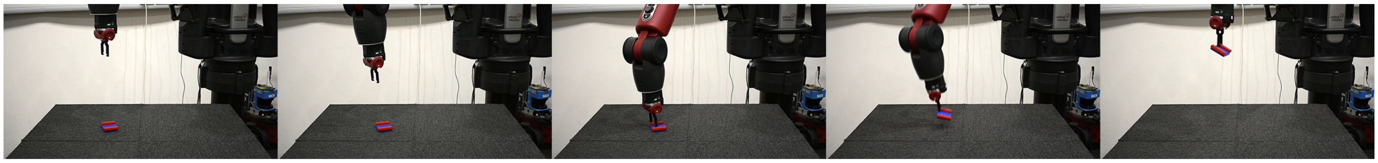
In the assembly task, we used the same Lego blocks as described in the recognition and picking experiments the Red Lego block and the RedBlue Lego block. The Lego block has multi pegs on one side and multi holes on the opposite side. Therefore, the assembly task is the Peg-in-Hole task, i.e., insert the pegs into the holes. Figure 6a shows the human demonstrator teaches the robot to pick up the RedBlue Lego block from the "components zone" to the "assembly zone" by kinesthetic guiding. Then the demonstrator teaches the robot to pick up another object (Red Lego block) from the "components zone" and moves to the "assembly zone", finally assemble the two objects, as shown in Figure 6b.

To validate that the robot was able to assemble by itself, we placed the RedBlue block under the wrist camera within the "components zone". The robot inferred a good grasping pose and grasped the RedBlue block successfully. After the robot placed the RedBlue block in the "assembly zone", we placed the second workpiece, the Red block, at a random location. The robot found a successful grasping pose and assembled the two blocks successfully at last (see Figure 6c).

It should be noted that the assembly movement is controlled by a force threshold. When the robot is executing the assembly movement, the force increases gradually until it reaches the threshold. The threshold is manually adjusted according to

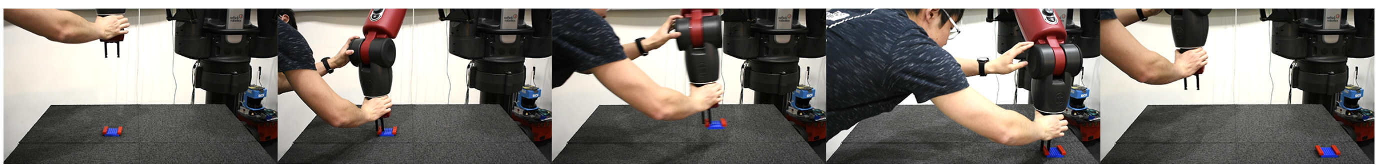


(a)



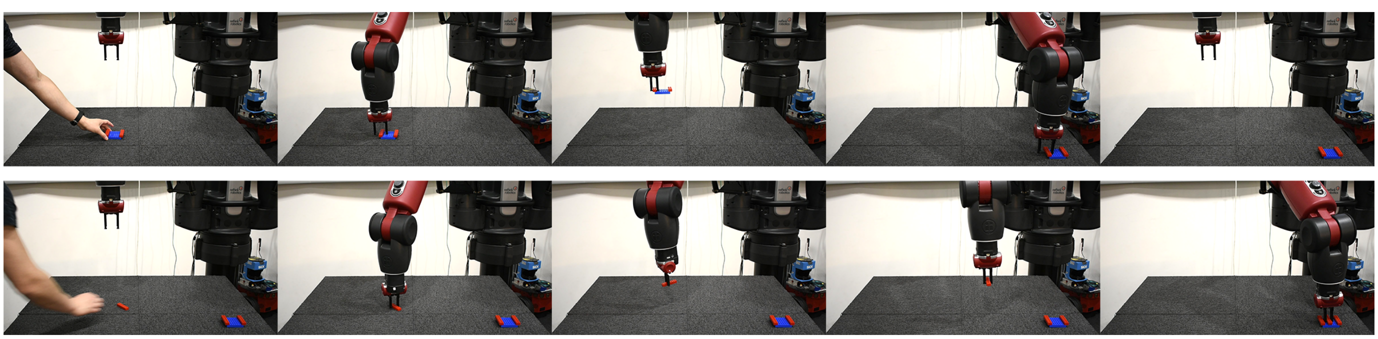
(b)

Fig. 5: Robot learns the skill of picking. 5a) Human demonstrates how to pick a Lego block from an effective position and orientation by kinesthetic guiding. 5b) Robot reproduces picking skill with the learned object in arbitrary positions.



(a)

(b)



(c)

Fig. 6: Robot learns the skill of assembly. 6a) The human demonstrator teaches the robot to pick the first workpiece to the assembly location by kinesthetic guiding, waiting for the following assembly steps. 6b) The human demonstrator teaches the robot to pick the second workpiece to the assembly location and assemble the second workpiece into the first workpiece. 6c) The robot reproduces the assembly task autonomously.

experience data. In this paper, the threshold is set at $14N$. In the teaching process, the human demonstrator taught the robot the grasping point and orientation, as well as the assembly sequence. It speeded up the learning progress of robotic assembly.

V. Conclusion

In this paper, we proposed a new method for learning grasping pose used in an assembly task. Kinesthetic guiding is used for the learning. Force control is implemented for controlling assembly movement. The key target was to simplify the teaching process of the assembly task.

Experiments from the Lego Blocks assembly task show that the proposed method can be used in teaching robots to do assembly tasks through simple demonstrations. However, further experiments are needed to study the robustness of the system over different assembly tasks, such as slide-in-the-groove, bolt screwing, and finally chair assembly.

In the future, we will extend the single arm manipulation to dual-arm manipulation. The additional arm and wrist camera enable the transfer of more assembly skills to robots. During the assembly phase, the force control strategy needs to be optimized to ensure a smooth motion and correct the assembly positions.

Acknowledgment

Zuyuan Zhu is financially supported by China Scholarship Council and the University of Essex Scholarship for his PhD study.

References

[1] S. Hu, J. Ko, L. Weyand, H. ElMaraghy, T. Lien, Y. Koren, H. Bley, G. Chryssolouris, N. Nasr, and M. Shpitalni, "Assembly system design and operations for product variety," *CIRP Annals*, vol. 60, no. 2, pp. 715 – 733, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000785061100206X

[2] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus, "Ikeabot: An autonomous multi-robot coordinated furniture assembly system," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, Conference Proceedings, pp. 855–862.

[3] F. Surez-Ruiz and Q.-C. Pham, "A framework for fine robotic assembly," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, Conference Proceedings, pp. 421–426.

[4] N. Krger, A. Ude, H. G. Petersen, B. Nemec, L.-P. Ellekilde, T. R. Savarimuthu, J. A. Rytz, K. Fischer, A. G. Buch, and D. Kraft, "Technologies for the fast set-up of automated assembly processes," *KI-Knstliche Intelligenz*, vol. 28, no. 4, pp. 305–313, 2014.

[5] Z. Zhu and H. Hu, "Robot learning from demonstration in robotic assembly: A survey," *Robotics*, vol. 7, no. 2, p. 17, apr 2018.

[6] Y. Yang, L. Lin, Y. Song, B. Nemec, A. Ude, A. G. Buch, N. Krger, and T. R. Savarimuthu, "Fast programming of peg-in-hole actions by human demonstration," in *Mechatronics and Control (ICMC), 2014 International Conference on*. IEEE, 2014, Conference Proceedings, pp. 990–995.

[7] Y.-L. Kim, H.-C. Song, and J.-B. Song, "Hole detection algorithm for chamferless square peg-in-hole based on shape recognition using f/t sensor," *International journal of precision engineering and manufacturing*, vol. 15, no. 3, pp. 425–432, 2014.

[8] K. Nottensteiner, M. Sagardia, A. Stemmer, and C. Borst, "Narrow passage sampling in the observation of robotic assembly tasks," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, Conference Proceedings, pp. 130–137.

[9] B. Nemec, F. J. Abu-Dakka, B. Ridge, A. Ude, J. A. Jorgensen, T. R. Savarimuthu, J. Jouffroy, H. G. Petersen, and N. Kruger, "Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile," in *Advanced Robotics (ICAR), 2013 16th International Conference on*. IEEE, 2013, Conference Proceedings, pp. 1–7.

[10] T. Tang, H.-C. Lin, Y. Zhao, Y. Fan, W. Chen, and M. Tomizuka, "Teach industrial robots peg-hole-insertion by human demonstration," in *Advanced Intelligent Mechatronics (AIM), 2016 IEEE International Conference on*. IEEE, 2016, Conference Proceedings, pp. 488–494.

[11] A. Kramberger, R. Piltaver, B. Nemec, M. Gams, and A. Ude, "Learning of assembly constraints by demonstration and active exploration," *Industrial Robot: An International Journal*, vol. 43, no. 5, pp. 524–534, 2016.

[12] R. Schmitt and Y. Cai, "Recognition of dynamic environments for robotic assembly on moving workpieces," *The International Journal of Advanced Manufacturing Technology*, vol. 71, no. 5-8, pp. 1359–1369, 2014.

[13] A. Sari, J. Xiao, and J. Shi, "Reducing uncertainty in robotic surface assembly tasks based on contact information," in *Advanced Robotics and its Social Impacts (ARSO), 2014 IEEE Workshop on*. IEEE, 2014, Conference Proceedings, pp. 94–100.

[14] *Ein*, Humans To Robots Laboratory. [Online]. Available: http://h2r.cs.brown.edu