2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Automated Assembly Skill Acquisition through Human Demonstration

Ye Gu, Weihua Sheng, and Yongsheng Ou

Abstract— **Acquiring robot assembly skills through human demonstration is a challenging problem. To achieve this goal, not only the actions and objects have to be shown to the robot, but also the effect of the action needs to be estimated. Recognizing the subtle assembly actions is a non-trivial task, and it is difficult to estimate the effect of the action on the assembly parts due to the small part sizes. In this paper, with a RGB-D camera, we build a Portable Assembly Demonstration (PAD) system which can recognize the part/tool used, the action applied and the assembly state characterizing the spatial relationship between the parts. The experiment results proved that this PAD system can generate an assembly script with good accuracy in object and action recognition as well as assembly state estimation.**

I. INTRODUCTION

A. Motivation

As more and more advanced robots enter our life, the demand for teaching robot complex skills increases. If a robot can replace a human doing complex assembly tasks, the expense of labor force can be reduced dramatically. For example, Foxconn Technology Group has already deployed their own assembly robot called "Foxbot" in their factories [1]. Currently these robots can only perform simple and repetitive tasks like picking and placing a part. Many of the complicated processes still require human labors. Teaching a robot such delicate assembly skills through human demonstration can avoid lengthy robot programming, while no technical expertise is required for the operator. However, a complex assembly task usually involves many parts and tools and is conducted in many steps, which introduces some challenges. First, recognizing small parts and tools using computer vision is not an easy task. Second, it is hard to recognize fine actions using motion based features only. Third, to obtain the relative position and orientation information between the parts is challenging, since occlusion frequently occurs between assembled parts, which may affect the efficiency of traditional object pose estimation algorithms. Finally, to capture the information of both the objects (parts and tools) and human movement, traditional approaches use multiple sophisticated sensors, which is costly and not easy to use.

Existing human based demonstration systems focus on extracting either human motion information or object information from the demonstration. In [2] and [3], object information is extracted from the demonstration to create high-level knowledge. In [4] humans are tracked by multiple calibrated stereo cameras for human motion imitation. However, for assembly skill learning, both the objects and human motion in the demonstration are important clues. The human motion can tell how to do a task while the object information can tell what parts are being manipulated or what tools are being used. Dillmann [5] built a human-based demonstration platform using data gloves to capture human motion while using multiple cameras for object recognition. On the contrary, we adopt a single RGB-D camera to develop a Portable Assembly Demonstration (PAD) system. This PAD system can capture the information about human motion, the tools used, and the parts manipulated during human demonstration to generate the skill scripts. The features of the PAD system are as follows. First, it can recognize small assembly parts and tools used. Second, with the parts and tools as the contextual information, the assembly action can be recognized. Third, the final state (post-condition) of the assembled parts can be estimated which represents the effect of the action. The state describes the relative position and orientation between the parts. With the PAD system, for example, it is possible to estimate how deep a bolt is hammered into a hole in the demonstration, which is important for the robotic assembly process.

B. Related work

The ways to demonstrate skills to robots fall into two categories: robot-based demonstration and human-based demonstration. Robot-based demonstration uses robots as the demonstrator. This method mainly includes teleoperation [6] and kinesthetic teaching which is also called scaffolding [7]. On the contrary, human-based demonstration employs humans as the demonstrator. Compared to robot-based demonstration, human-based demonstration is more convenient for the operators since they do not need to control the robot. Two kinds of sensors are widely used for collecting the human demonstration data: wearable sensors and vision sensors. Wearable sensors include inertial measurement units (IMUs) [8] and data gloves [5], [9]. In robot learning, it is also called the "sensor on teacher" approach. Wearable sensors are usually used along with vision sensors that extract the information of the involved objects. On the other hand, vision-based human motion tracking systems are regarded as a natural way to capture demonstration since no sensors need to be attached to the demonstrator. There have been some works on vision-based assembly skill learning from human demonstration. Dantam *et al.* [10] developed a method to transfer human demonstration to robots. The demonstration

This project is supported by NSF grant CISE/IIS 1231671.

Ye Gu and Weihua Sheng (contacting author) are with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA e-mail: weihua.sheng@okstate.edu.

Yongsheng Ou is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong 518055, China and The Chinese University of Hong Kong, China

is interpreted as a sequence of object connection symbols which can be further transformed into the task language. The assembly task they handled is simple and only the location of the part is required. Takamatsu *et al.* [2] aimed to recognize assembly tasks through human demonstration. Two rigid polyhedral objects are recognized from noisy data obtained by a conventional 6 degree-of-freedom (DOF) object-tracking system. Assembly tasks can be basically expressed as chains of two-object relationships. Aleotti *et al.* [3] demonstrated the assembly task through simulation, which does not address the action recognition and object recognition problems. A task planner is designed to analyze the demonstration and segment it into a sequence of action primitives.

In this paper, we aim to develop a single Kinect based human skill demonstration system which recognizes the parts, the tools, the assembly actions and the assembly state. The rest of the paper is organized as follows: Section II introduces the PAD system setup and formulates the problem to be solved. Section III presents the proposed methodology. Section IV describes the experimental procedures and gives the experimental results. In Section V, the conclusion is drawn and future work is proposed.

II. System overview

A. PAD System Setup

The PAD system is shown in Fig. 1. A single Kinect sensor is used to capture the information about the objects and the human motion. The PAD system has three operation modes: object modeling mode, object recognition mode and human tracking mode. To model and recognize objects, the Kinect faces downward 45 degree at the objects. To track human skeleton, the Kinect looks horizontally at the demonstrator. The pose of the Kinect can be controlled through the motor embedded in it. The 3D modeling subsystem, or 3D scanner, consists of a rotational Lazy Susan and a Kinect sensor. To create the 3D model of an object, we adopt the RoboEarth package [11]. Instead of manually rotating the marker template, we attach the marker template to the Lazy Susan and use a step motor to control its rotation. The 3D model can be built after the Lazy Susan rotates 360° at a proper speed.

Before the skill demonstration, the 3D models for all the parts and tools will be created. The 3D models of the tools will be used for tool recognition while the 3D models of the parts will be used for part recognition and assembly state estimation. The procedure for the demonstration is as follows. First, the tools and the parts used in the assembly will be put on the 3D scanner for recognition. Second, the parts will be assembled through certain actions, possibly with the help of a tool. Third, the assembled part will be put on the 3D scanner for modeling and assembly state estimation. This process is repeated until all the parts are assembled into the final product.

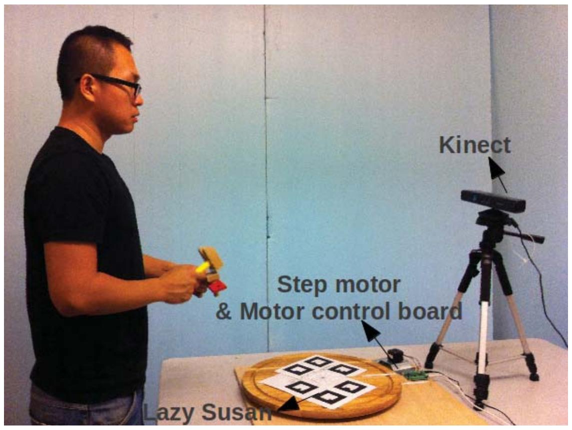


Fig. 1. The hardware setup of the PAD system.

B. Formulation of the problem

Generally a complex assembly task which involves n parts can be represented as a sequence of subtasks as follows:

$$final\ product = \underbrace{\underbrace{\underbrace{p_1}_{{M_1}^1} \oplus \underbrace{p_2}_{{M_2}^1}}_{{M_1}^2} \oplus \underbrace{p_3}_{{M_2}^2} \oplus ...}_{{M_1}^{n-1}} \oplus \underbrace{p_n}_{M_2^{n-1}} \qquad (1)$$

where p_n is the nth part. $\oplus$ is the assembly action on the two parts. It can be considered as two-part assembly tasks recursively. In a two-part assembly task, the models for each part are defined as M_1 and M_2. First, take p_1's model as ${M_1}^1$, p_2's model as ${M_2}^1$. The superscript denotes the index of the subtask. Then, ${M_1}^2$ and ${M_2}^2$ will be used to denote $p_1 \oplus p_2$ and p_3 respectively. This will continue until the last part. In this work, we only consider the two parts case.

Given the above setup, two problems need to be solved. The first is to recognize the action and all the parts and tools involved, the second is to estimate the assembly state which represents the effect of the action.

The input to the action recognition system has two parts: the observation of the objects and the observation of the human action. The observation of the objects has two parts: ψ_p is the decision of the part recognition which is one of the parts in the assembly sets. ψ_t is the decision of the tool recognition which is one of the tools in the tool set. If the action does not involve any tool then $\psi_t = null$. The observation of the action ψ_a is a sequence of skeleton data of the human subject $\{\Gamma_1, \Gamma_2 ... \Gamma_t\}$, where $\Gamma_t = \{\Theta_1, \Theta_2 ... \Theta_k\}$, Θ_i is the angle of joint i and k is the number of joints. We assume the part set $P = \{P_1, P_2 ... P_m\}$, the tool set $T = \{T_1, T_2 ... T_n\}$, and the action set $A = \{A_1, A_2 ... A_q\}$. The problem is to develop an algorithm to decide on $A_i \in A$ given the above observation. In other words, the goal of the action recognition is to find a mapping function f_1,

$$A_i = f_1(\psi_a, \psi_p, \psi_t) \qquad (2)$$

To obtain the assembly state, the input includes the 3D models of each individual part and the 3D model of the assembled part after each action. The goal of the assembly state estimation is to find a mapping function f_2

$$S = f_2(M_1, M_2, M_1 \oplus M_2) \tag{3}$$

Here, S is assembly state. M_1 and M_2 are the 3D models of the two parts involved. $M_1 \oplus M_2$ is the 3D model of the assembled part which includes parts M_1 and M_2. For state estimation, M_2 is treated as the reference part. Then the state S is the pose of the non-reference part with respect to the reference part.

Based on the solutions to the two problems mentioned, we can generate a sequence of skill scripts Σ as follows

$$\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3 ... \Sigma_n) \tag{4}$$

where each skill script Σ_i is a 5-tuple symbolic description defined as

$$\Sigma_i =< A_i, T_i, P_{1i}, P_{2i}, S_i > \tag{5}$$

III. Methodology

In this section, the approaches to object recognition, action recognition and assembly state estimation are introduced.

A. Object recognition

With the emergence of RGB-D cameras, color and depth based object recognition has been receiving more attention recently [12]. With RGB-D cameras, not only the object type but also its location and orientation can be obtained efficiently, which is helpful for object manipulation. By recognizing the parts to be assembled and the tool to be used, we can have better recognition of the assembly action. The challenge is that the size of the parts and tools are usually small and traditional 2D feature based recognition algorithms are not effective due to the limited number of features. Therefore we adopt a 3D feature based recognition algorithm.

We first create the 3D object models based on the 3D scanner we described. Then 3D object recognition is performed based on the recognition module in Point Cloud Library (PCL) [13]. First, the normals of both the model and the scene clouds are computed using the 10 nearest neighbors of each point. Second, each point cloud is downsampled in order to find a small number of keypoints, which will then be associated to a 3D descriptor in order to perform keypoint matching and determine point-to-point correspondences. Third, point-to-point correspondences between model descriptors and scene descriptors are determined. The last stage is to cluster the previously found correspondences. Two correspondence grouping algorithms [14], [15] are used. For more details about the object recognition, please see [13].

B. Action recognition

We aim to build an accurate action recognition system by considering the object/action correlation. A two level probabilistic approach is proposed to achieve this goal. At the low level, Hidden Markov Models (HMMs) [16] are employed to model the dynamics of the actions. At the high level, a Bayesian model is used to capture action-object dependencies. The framework is shown in Fig. 2.

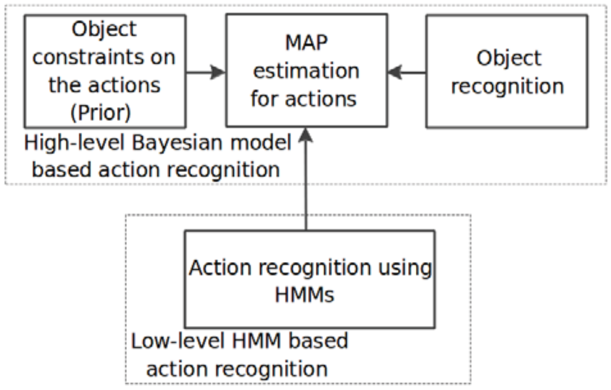

Object constraints on the actions (Prior)
MAP estimation for actions
Object recognition
High-level Bayesian model based action recognition
Action recognition using HMMs
Low-level HMM based action recognition

Fig. 2. The action recognition framework.

1) HMMs for low-level action recognition: Modeling the low-level dynamics of human motion is important for human motion recognition. It serves as a quantitative representation of simple movements so that those simple movements can be recognized in a reduced space by the trajectories of motion parameters [17]. We propose a single-step HMM based approach to perform real-time action spotting and classification of continuous user motion.

The raw right arm joint angles can be accessed with the support of the Openni driver [18] and the skeleton tracker function as shown in Fig. 3. After segmentation and symbolization, the data can be fed into HMMs for training and recognition. For the low-level HMMs, each model is trained with fifteen sets of training data at a sampling rate of 20 Hz. A rule-based method is adopted for data segmentation. The starting pose is defined, and each training data set consists of twenty data points after the starting point. Then K-means clustering algorithm is used to convert the vectors into the observable symbols for HMMs. The centroids from K-means are saved for use in testing. To balance the computational complexity, efficiency and accuracy, we set up parameters for each HMM as follows: the number of states in the model is 10; the number of distinct observation symbols is 8. For further details about the low-level HMM please see our previous work [19].

2) Bayesian network for modeling object action dependencies: There are two kinds of objects, the tools and the parts. They both have correlation with the action. Manipulating action is denoted by A_j. ψ_p is the decision of the part recognition. ψ_t is the decision of the tool recognition. ψ_a is the observation of the action, where $\psi_p \in \Psi_p$, $\psi_t \in \Psi_t$ and $\psi_a \in \Psi_a$. At this level, the goal is to find the maximum posterior likelihood (MAP) estimation, $\underset{A_j}{max} P(A_j|\psi_a, \psi_p, \psi_t)$. According to Bayesian rule,

$$P(A_j|\psi_a, \psi_p, \psi_t) \propto P(\psi_a|A_j, \psi_p, \psi_t) \cdot P(A_j|\psi_p, \psi_t) \tag{6}$$

$P(\psi_a|A_j, \psi_p, \psi_t) = P(\psi_a|A_j)$, because according to the

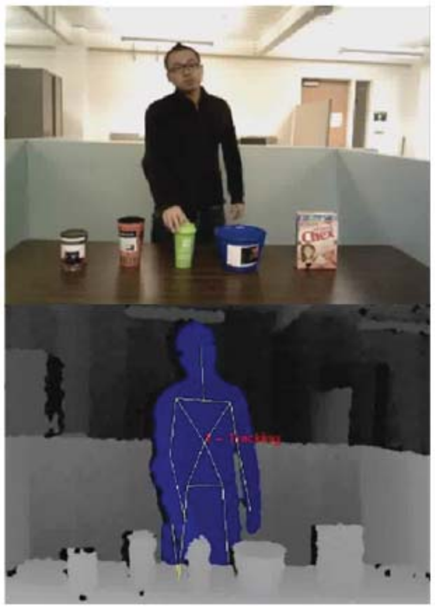
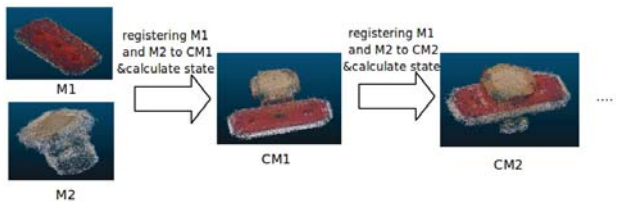


Fig. 3. Human skeleton tracking using a Kinect sensor.


registering M1 and M2 to CM1 &calculate state
registering M1 and M2 to CM2 &calculate state
M1
M2
CM1
CM2
....
Fig. 4. State estimation for a two-step assembly task. M_1 and M_1 are the 3D model of the base board and the bolt. CM_1 is the 3D model after the first action. CM_2 is the 3D model after the second action.

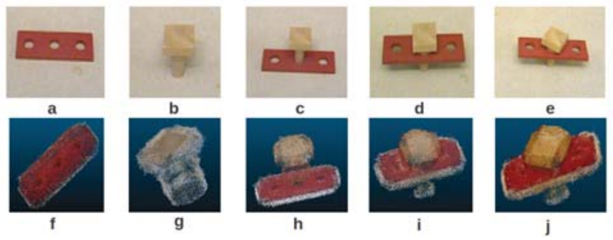


Fig. 5. The parts and the associated model. The images on top show the parts. (a) base board, (b) bolt, (c) after "insertion", (d) after "hammering". (e) after "wrenching". The images on bottom shows the corresponding 3D models (f) M_1. (g) M_2. (h) CM_1. (i) CM_2. (j) CM_3.

Bayesian network, ψ_a is independent of ψ_p and ψ_t given A_j. $P(\psi_a|A_j)$ is the action recognition output in terms of probability from the low-level HMMs. On the other hand, according to the Bayesian network, $P(A_j|\psi_p,\psi_t) \propto P(A_j|\psi_p)\cdot P(A_j|\psi_t)$. Applying the total probability theorem, we have

$$P(A_j|\psi_p) = \sum_i P(A_j|P_i,\psi_p)\cdot P(P_i|\psi_p) \tag{7}$$

$$P(A_j|\psi_t) = \sum_i P(A_j|T_i,\psi_t)\cdot P(T_i|\psi_t) \tag{8}$$

If part P_i and tool T_i are not used, the dominant factor in Equation (7) and (8) are $P(A_j|P_s,\psi_p)$ and $P(A_j|T_s,\psi_p)$ where P_s and T_s indicate the involved part and the tool respectively. $P(A_j|P_s,\psi_p) = P(A_j|P_s)$, since A_j and ψ_p are independent given P_s. $P(A_j|P_s)$ is the prior probability characterizing the action that is applied on this part, which can be calculated based on the occurrence of A_j given P_s in the training set. On the other hand, $P(P_s|\psi_p)$ is the part classification accuracy. Similarly, $P(A_j|T_s,\psi_t) = P(A_j|T_s)$, $P(A_j|T_s)$ is the prior probability characterizing the action that occurs when this tool is used, which can be calculated based on the occurrence of A_j given T_s in the training set. $P(T_s|\psi_t)$ is the tool classification accuracy. Therefore, we have

$$\begin{aligned}&P(A_j|\psi_p,\psi_t) = \\ &P(A_j|P_s)\cdot P(P_s|\psi_p)\cdot P(A_j|T_s)\cdot P(T_s|\psi_t)\end{aligned} \tag{9}$$

If the action does not involve any tool, we have

$$P(A_j|\psi_p) = P(A_j|P_s)\cdot P(P_s|\psi_p) \tag{10}$$

C. Assembly state estimation

All the models have the same coordinate frame defined by the 3D scanner, since all the parts are scanned using the same camera pose. One of the two parts will be treated as the reference part. Therefore the state is the pose of the non-reference part with respect to the reference part.

Fig. 4 shows a two-step assembly task that inserts and hammers a square bolt into a hole on the base board. With the 3D scanner, first we create the 3D models for these two parts. Since they have the same frame, the initial state S_0 can be represented as a 4 by 4 identity matrix. The 3D model for the assembled part is created once the action is applied to these two parts. The 3D models for each part will be registered to the 3D model of the assembled part respectively. Iterative Closest Point (ICP) [20] algorithm is adopted to calculate the transformation.

We obtain two transformation matrices. T_{ri} represents the pose change of the reference part after registration; while T_{pi} represents the pose change of the non-reference part after registration. The assembly state S_i after action i can be calculated as.

$$S_i = (T_{r(i-1)})^{-1}\cdot S_{i-1}\cdot T_{p(i-1)} \tag{11}$$

IV. Experiment and results

In this section, the design of the experiments is introduced first, then the experimental results are presented.

A. Experiment procedure

We use some toy parts and tools to conduct the assembly task. The parts are a bolt and a base board with holes. The tools include a screw driver, a hammer and a wrench. The associated actions are screwing, hammering and wrenching.

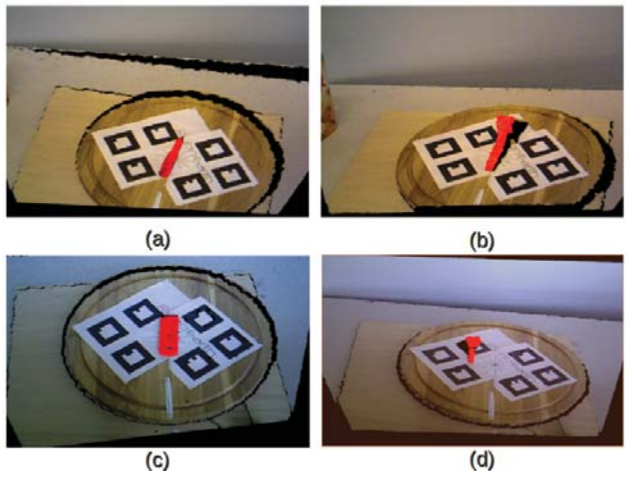


Fig. 6. Results of tool and part recognition, where the recognized objects, the recognized objects are highlighted in red. (a) recognition of the screw driver. (b) recognition of the hammer. (c) recognition of the base board. (d) recognition of the screw.

In the first experiment, a bolt is inserted and then hammered into a hole on a base board. Then, it is wrenched by 30° with respect to the base board. Fig. 5 shows the parts and the corresponding 3D models. The second experiment is to find an appropriate point cloud downsampling rate to accelerate the ICP computation without sacrificing the performance. The bolt is wrenched around the Z axis with a step of 30° from 0° to 180°. At each step, we create a 3D model for the assembled part. Then the state S after each wrenching action is estimated. Due to the small size of the part, the scanner can not capture the details very clearly. However, it does not affect the registration results, since only the outline of each point cloud matters.

B. Experiment Results

1) Part recognition: The top row in Fig. 6 shows the recognition of the screw driver and the hammer respectively. In the bottom row, it shows the recognition of the base board and the screw. The recognized objects are highlighted in red. In order to recognize small objects like the screw which has small size in the scene, the down sampling radius for the scene has to be small so that enough keypoints can be extracted. The average recognition accuracy for the parts and tools is above 85%.

2) Action recognition: For action recognition, we compare the low-level HMM output with the Bayesian model output. Based on the motion features only, the recognition accuracy is poor, since the motion features of these three actions are very similar to each other. With the object context, the actions can be differentiated effectively. To evaluate the action recognition system, each action is conducted 50 times. The accuracy of these two models are shown in Table I and Table II respectively.

3) Assembly state estimation: The state can be represented by a transformation matrix, from which the position and orientation can be calculated as $\{D_x, D_y, D_z, \psi, \theta, \phi\}$,

TABLE I

ACTION RECOGNITION ACCURACY OF THE LOW-LEVEL HMMS

test type	decision type				accuracy
	hammering	screwing	wrenching	not detected	
hammering	**0.44**	0.31	0.21	0.04	**0.44**
screwing	0.25	**0.46**	0.22	0.07	**0.46**
wrenching	0.17	0.28	**0.52**	0.03	**0.52**

TABLE II

ACTION RECOGNITION ACCURACY OF THE BAYESIAN MODEL

test type	decision type				accuracy
	hammering	screwing	wrenching	not detected	
hammering	**0.95**	0	0	0.05	**0.95**
screwing	0.04	**0.92**	0	0.04	**0.92**
wrenching	0.05	0	**0.91**	0.04	**0.91**

where D_x, D_y, D_y are the x, y, z positions and ψ, θ, ϕ are the roll, pitch and yaw angles. The initial state S_0 is

$$S_0 = \{0m, 0m, 0m, 0^{\circ}, 0^{\circ}, 0^{\circ}\} \tag{12}$$

Fig. 7 gives the results of registering M_1 and M_2 to CM_1 after the "insertion" action. The state S_1 which represents the pose of M_1 with respect to M_2 can be estimated as:

$$\hat{S}_1 = \{0.0010m, 0.0028m, -0.0029m, 0.00^{\circ}, 0.00^{\circ}, 0.01^{\circ}\} \tag{13}$$

The result shows that $\hat{S}_1$ is almost the same as S_0. It is because the pose of each individual model almost overlaps with its counterpart in CM_1. Therefore, T_{p1} and T_{r1} are both close to an identity matrix. Fig. 8(a) and 8(b) show the result of the registration of M_1 and M_2 in CM_1 to their poses in CM_2. The movement of M_1 is mainly translation along the vertical axis (Z axis). The ground truth of S_2 is $\{0m, 0m, 0.034m, 0^{\circ}, 0^{\circ}, 0^{\circ}\}$. The estimated S_2 is shown below

$$\hat{S}_2 = \{-0.0014m, 0.0031m, 0.0350m, 0.95^{\circ}, -0.74^{\circ}, 1.41^{\circ}\} \tag{14}$$

After wrenching, the bolt is rotated anticlockwise by 30° about the base board. Fig. 8(c) and 8(d) show the registration result of M_1 and M_2 in CM_2 to CM_3. The ground truth of S_3 is $\{0m, 0m, 0.034mm, 0^{\circ}, 0^{\circ}, 30^{\circ}\}$. The estimated S_3 is

$$\hat{S}_3 = \{0.0023m, -0.0011m, 0.0350m, 2.21^{\circ}, 1.36^{\circ}, 31.47^{\circ}\} \tag{15}$$

This assembly process is repeated 10 times. The average error for translation and rotation is within $5\ mm$ and 5° respectively. Finally, after the human demonstration, the following script can be generated automatically:

$$\begin{cases} \Sigma_1 = < insertion, null, bolt, baseboard, S_1 > \\ \Sigma_2 = < hammering, hammer, bolt, baseboard, S_2 > \\ \Sigma_3 = < wrenching, wrench, bolt, baseboard, S_3 > \end{cases} \tag{16}$$

V. CONCLUSION

In this paper, we propose a PAD system for automated assembly skill acquisition. With the recognition of the tools and parts used, we can effectively recognize assembly actions. To estimate the assembly state, we build 3D models of

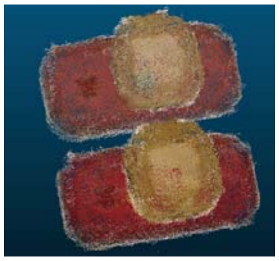


Fig. 7. Registration of the parts after "insertion". 3D model of the assembled part CM_1 is shown on top. The model on bottom is composed of the M_1 and M_2 after registering to CM_1.

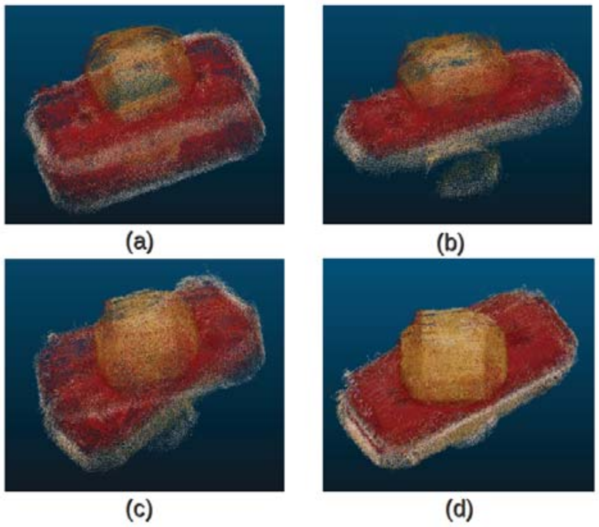


Fig. 8. Registration of the parts after "hammering" and "wrenching". (a) poses of M_1, M_2 in CM_1 and CM_2. (b) result of registering M_1, M_2 to CM_2. (c) poses of M_1, M_2 and CM_3. (d) result of registering M_1, M_2 in CM_2 to CM_3.

the individual parts and the assembled parts. ICP algorithm is used to obtain the spatial relationship between the parts. This method is robust to minor occlusions and works for small parts. Our PAD system can be used for robots to acquire assembly skills. The limited accuracy of the created 3D model limits the current PAD system to only simple parts. In future work, we will explore the use of more accurate 3D modeling methods and also try on more complicated parts.

REFERENCES

[1] Robohub. [Online]. Available: http://robohub.org/foxbots-being-deployed-in-china/

[2] J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi, "Recognizing assembly tasks through human demonstration," *Int. J. Rob. Res.*, vol. 26, no. 7, pp. 641–659, July 2007.

[3] J. Aleotti, S. Caselli, and M. Reggiani, "Toward programming of assembly tasks by demonstration in virtual environments," in *IEEE International Workshop on Robot and Human Interactive Communication*, 2003, pp. 309–314.

[4] P. Azad, T. Asfour, and R. Dillmann, "Toward an unified representation for imitation of human motion on humanoids," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 2558–2563.

[5] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 109–116, 2004.

[6] H. Friedrich, S. Münch, R. Dillmann, S. Bocionek, and M. Sassin, "Robot programming by demonstration (rpd): Supporting the induction by human interaction," *Machine Learning*, vol. 23, no. 2-3, pp. 163–189, 1996.

[7] S. Calinon and A. Billard, "A probabilistic programming by demonstration framework handling constraints in joint space and task space," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 367–372.

[8] C. Zhu and W. Sheng, "Realtime human daily activity recognition through fusion of motion and location data," in *2010 IEEE International Conference on Information and Automation*, june 2010, pp. 846 –851.

[9] R. Zollner, O. Rogalla, and R. Dillmann, "Integration of tactile sensors in a programming by demonstration system," in *IEEE International Conference on Robotics and Automation*, vol. 3, 2001, pp. 2578–2583.

[10] N. Dantam, I. Essa, and M. Stilman, "Linguistic transfer of human assembly tasks to robots," in *International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 237–242.

[11] D. Di Marco, A. Koch, O. Zweigle, K. Haussermann, B. Schiessle, P. Levi, D. Galvez-Lopez, L. Riazuelo, J. Civera, J. Montiel, M. Tenorth, A. Perzylo, M. Waibel, and R. van de Molengraft, "Creating and using roboearth object models," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 3549 –3550.

[12] K. Lai, L. Bo, X. Ren, and D. Fox, "Detection-based Object Labeling in 3D Scenes," in *IEEE International Conference on on Robotics and Automation*, 2012.

[13] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[14] F. Tombari and L. Di Stefano, "Object recognition in 3d scenes with occlusions and clutter by hough voting," in *Fourth Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, 2010, pp. 349–355.

[15] H. Chen and B. Bhanu, "3d free-form object recognition in range images using local surface patches," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 3, 2004, pp. 136–139.

[16] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, 1989, pp. 257–286.

[17] Y. Wu, T. S. Huang, and N. Mathews, "Vision-based gesture recognition: A review," in *Lecture Notes in Computer Science*, pp. 103–115.

[18] Openni. [Online]. Available: http://www.openni.org/Documentation/

[19] Y. Gu, H. Do, J. Evert, and W. Sheng, "Human gesture recognition through a kinect sensor," in *IEEE International Conference on Robotics and Biomimetics*, Dec. 2012.

[20] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.