Assembly Motion Recognition Framework Using Only Images

Kosuke Fukuda[1], Natsuki Yamanobe[2], Ixchel G. Ramirez-Alpizar[2,1] and Kensuke Harada[1,2]

***Abstract*— This work proposes a method for recognizing and segmenting assembly tasks into single motions. First, using a motion capture system based on pose estimation from multiple points, we obtain a time series data of the human's motion during an assembly task (motion data). We use an object detector algorithm to determine the assembly parts and tools that the user (human) is grasping. Then, we divide (segment) the assembly motion based on the change of the manipulated object and the velocity of the hand. We carry out the motion recognition of the segmented motion data by using several Hidden Markov Models (HMMs) that represent the actions that can be executed with the manipulated object(s). We recorded the assembly motion of an airplane toy done by two experts for training the HMMs and recorded the assembly motion of five subjects to verify the validity of the proposed method.**

I. Introduction

In recent years, the number of ongoing research on the automatic robot motion generation of assembly tasks towards the automation of assembly tasks in factories has increased. Also, there has been research endeavors to construct a database to store human's motions for the automatic generation of assembly motions for robots [1], [2]. In these endeavors, the segmentation of an assembly task motion into single motions and its respective label (motion name) has to be done for each of these motions, in order to be stored in the database. So far, this segmentation and labeling is manually done and it becomes time consuming and tedious when we have a large-scale database. To solve this problem, in this work we propose a framework for the automatic motion segmentation and recognition of assembly motions done by humans without burdening them.

Previous work on motion recognition had aimed at recognizing the human's pose or gestures. Mori et al. used a Support Vector Machine (SVM) to recognized the pose of a human. They also proposed a segmentation method using HMMs based on the probability of the recognized pose [3]. Aksoy et al. proposed a method for clustering motions based on the description of the relationship between position and contact of the hand and object [4]. However, in these work the targeted motions are simple manipulations, everyday motions, etc., that do not involve fine movements of the fingertips. They have not tackled assembly motions done by humans. In this work, we aim to recognize manipulation motions involving fine movements of the fingertips such as screwing a bolt, using a screwdriver, etc.

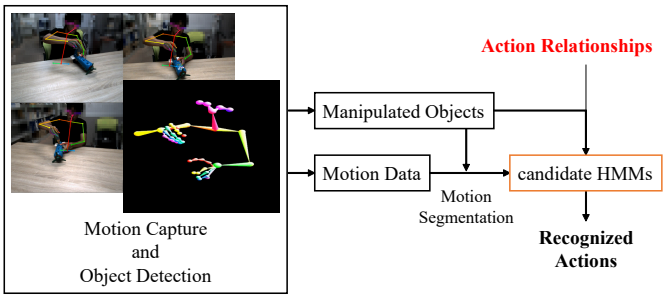


Fig. 1: Overview of proposed recognition system.

Regarding the identification of hand's motions and/or posture of fine manipulations, research using wearable sensors has been done. Kubota et al. used a surface electromyographic (sEMG) sensor in combination with a motion capture system and proved that the sEMG sensor is effective in identifying fine motions [5]. Adrien et al. recognized motions using information from datagloves, motion capture suits, etc., and analyzed the effectiveness of sensor information by performing a feature selection [6]. Even though using wearable sensors might improve the motion recognition performance, the reality is that for recording data at factories, etc., there are constraints in space and the environment thus it is almost impossible to use such sensors. For this reason, in this work, we only use four cameras and no special sensors for tracking the human's motions and the manipulated objects, as shown in Fig. 1. From the obtained data, we derive features equivalent to those obtained from wearable sensors and perform motion recognition.

As assembly tasks are composed of advanced manipulations involving fine motions of the fingertips, motions might look similar but depending on the manipulated object, purpose of the motion, etc., their name labels might differ. When the aim is to build a motion database for storing human's motions to be use by robots, it is very important to appropriately label these motions. In our previous work [7], we proposed a method for motion recognition that uses the concept of affordances to define action relationships with the manipulated objects. However, in this work we propose a method that does not use wearable sensors nor markers to track the human's and object's position, thus removing motion and space constraints generated by these sensors and markers. Also, we revise the segmentation method and improve it to make possible the recognition of assembly motions when the same object is being manipulated con-

[1]Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama-cho, Toyonaka, 560-8531, Japan {fukuda, harada}@hlab.sys.es.osaka-u.ac.jp

[2]Automation Research Team, Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology (AIST), 2-4-7 Aomi, Koto-ku, Tokyo, 135-0064, Japan {n-yamanobe, ixchel-ramirezalpizar}@aist.go.jp

tinuously. Furthermore, to prove the validity of the proposed framework and the role of the defined action relationships, we recorded the assembly task of an airplane toy done by five subjects.

This paper is organized as follows: in section II we describe the proposed motion recognition framework. Then, in section III we present, analyze and discuss the experimental results obtained. Finally, in section IV we give the main conclusions of this work and discuss future directions.

II. Motion Recognition Framework

In this section, the proposed framework for motion segmentation and recognition is described. First, we explain the action relationship used for narrowing down the possible action to be recognized. Next, we describe how the human motion data was obtained. Then, we explain the method used to identify the manipulated object. Finally, we describe the motion segmentation and Hidden Markov Models used for motion recognition. The outline of the proposed framework is shown in Fig. 1.

A. Action Relationships

In our previous work [7], we defined action relationships between manipulated objects and actions based on the affordance concept. These action relationships refer to the assembly actions that can be executed depending on the combination of objects being manipulated. Using this information, the motion candidates for recognition can be narrowed down, as long as the manipulated object is known (recognized). Thus, the efficiency and performance of the recognition process is expected to improve. In this work, besides verifying the validity of the proposed recognition framework, we also compare our results with those without using the action relationship definition.

B. Motion Data

The position of a human doing an assembly task is obtained using OpenPose [8]–[11]. From the data obtained using OpenPose, we compute the desired features for motion recognition. OpenPose is an open software based on deep learning that detects the human skeleton in 2D from a given image. It can also detect the hand and fingers' joints in 2D (x, y). However, from one image we cannot obtain the depth information of the human position, we can only obtain the 2D position. As we want to recognize assembly motions such as pick, screw, etc., where the change in 3D position is important, we use the 2D information from four cameras to compute by triangulation the 3D position information of the human motion, as shown in Fig. 2.

The hand and finger's joint position $\vec{p}_j = (x, y, z)$ obtained from the measured 3D position are in the camera coordinate frame. This joint position is a feature that impairs the generalization of the motion recognition because its value varies depending on where the camera is installed and the position of the human with respect to the camera. Therefore, instead of using the joint's position directly, we

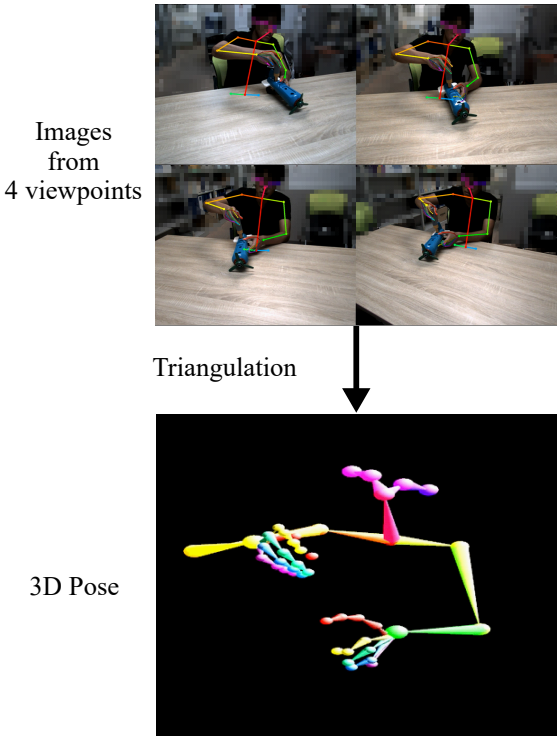


Fig. 2: Obtaining 3D pose from four 2D images.

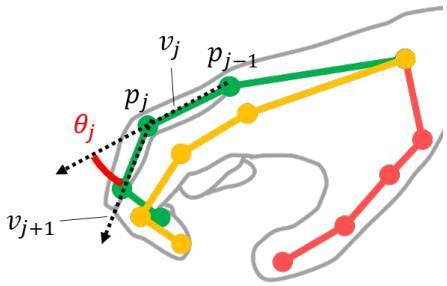


Fig. 3: Computation of joint's angles from joint's positions.

use the following features to preserve the generalization of our method.

- Joints angles of Thumb, Index, and Middle fingers
- Norm of the wrist velocity
- The average of the fingers' velocity norm
- Opening width of the fingers and its change

From the joint position $\vec{p}_j$ we compute the vector between joints as $\vec{v}_j = \vec{p}_j - \vec{p}_{j-1}$, from which we compute the joint angle:

$$\theta_j = \arccos \frac{\vec{v}_j \cdot \vec{v}_{j+1}}{\|\vec{v}_j\| \, \|\vec{v}_{j+1}\|},$$

as illustrated in Fig. 3. In case there is occlusion from the object, the 3D position of the human's hand skeleton is not obtained. In these cases, we do a linear interpolation between the last computed 3D positions. The velocity magnitude of the joint's position is computed as $\|\vec{p}_j(t) - \vec{p}_j(t-1)\|$, from which we compute the wrist velocity and the average velocity of the hand. These features related to the motion in the

workspace are use to represent motions like insert and hold, where the motion in space is more representative of the task than the joint's angles. The opening width of the fingers is defined by the distance between the middle and thumb fingers or the index and thumb fingers as: $opening\_width = \min\{\|\vec{p}_{\text{index}} - \vec{p}_{\text{thumb}}\|, \|\vec{p}_{\text{middle}} - \vec{p}_{\text{thumb}}\|\}$.

C. Object Detection and Identification of Manipulated Objects

In our previous work [7], we used AR markers to detect the position of the objects and used the distance between the hand and the object to determine the object that was being manipulated. However, the objects in which an AR marker can be attached are limited due to their size, thus, it is not possible to use AR markers when doing assembly motions where small parts are involved. Also, when using AR markers there is a limit on the usable working space due to the maximum recognizable distance of the AR marker and its possible occlusion throughout the task, resulting in unnatural motions that affect the motion recognition performance. For this reason, in this work we use the YOLO [12] object detector algorithm to obtain the object's position. The manipulated object is identified based on the human motion data obtained using the method described in section II-B. The YOLO object detector is based on a single Convolutional Neural Network (CNN) which makes it faster than other object detectors, and it also gives the user the option of trading off speed for accuracy. In this work, we use the class and bounding box information detected by YOLO.

The manipulated object identification is done following the next steps:

1) We employ the state transitions of a Hidden Markov Model (HMM) to differentiate hand states, as shown in Fig. 5. From the training data, all the motions are defined as "manipulate" except for the "hold", "pick" and "release" motions. "Release" is defined as the motion from the time the task finishes until 0.5 s. has passed. When the predicted state by the HMM is "manipulate" or "hold", the manipulated object is kept the same, otherwise we go to the next step. This step is intended to be robust against occlusion from the hand.
2) As shown in Fig. 4, we determine the manipulated object as the one that has inside its bounding box the 2D position of any of the fingers' joints (detected by OpenPose). In case there are more than two objects with fingers inside their bounding boxes, the object that has more joints inside its bounding box is chosen. Considering object detection failure, the last recorded data of the object detection is used for this step.

We obtain a time series of object data for each hand by identifying the manipulated objects. These data will be use for the motion segmentation (section II-D) and the motion recognition (section II-E).

D. Motion Segmentation

To carry out the motion recognition method described in section II-E it is necessary to segment the motion data into single motion segments. In our previous work [7] we only used the change in the manipulated object to segment the motion, however, the series of consecutive motions that do not change of manipulated object such as "pick", "insert", "screw" could not be correctly segmented. In this work, besides segmenting the motion based on the change of the manipulated object, we also segment based on hysteresis thresholding of the motion during the manipulation of one object. The intervals of time when then wrist velocity surpasses the first threshold $minVal$ and it also surpasses the second threshold $maxVal$ are segmented as "the arm is moving", otherwise it is segmented as "the arm is not moving", as shown in Fig. 6. In this paper, we define the thresholds empirically as minVal$= 0.05$ and maxVal$= 0.15$.

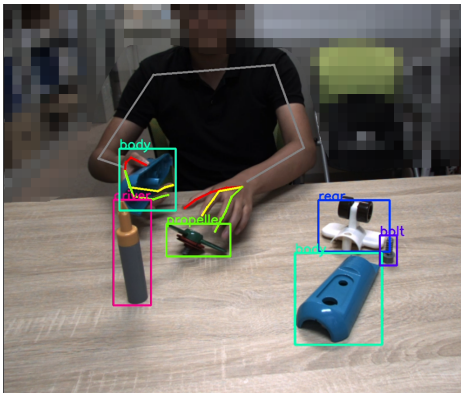


Fig. 4: Estimated 2D hand's position by OpenPose and detected objects by YOLO.

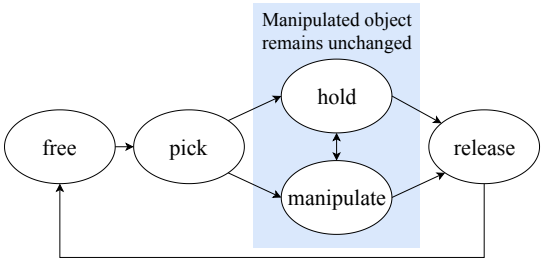


Fig. 5: HMM's states for identifying the manipulated object.

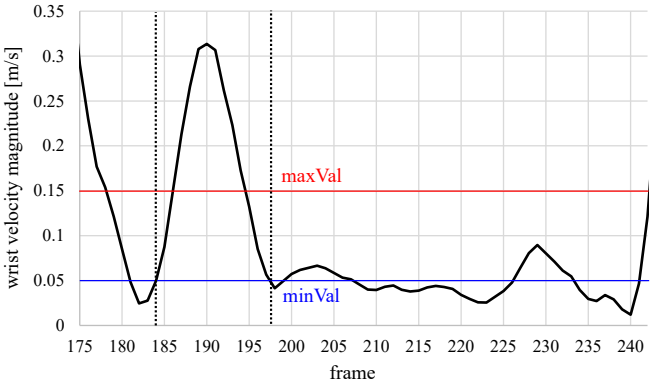


Fig. 6: Motion segmentation by thresholding using hysteresis on the wrist velocity.

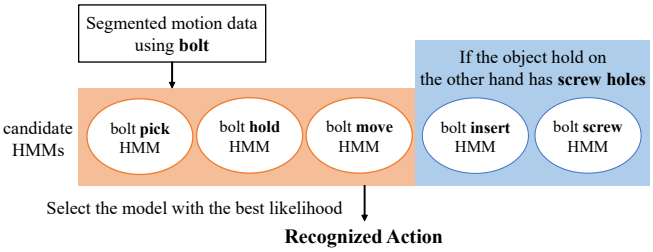


Fig. 7: Selection process of candidate HMMs from manipulated objects

E. Motion Recognition using HMMs

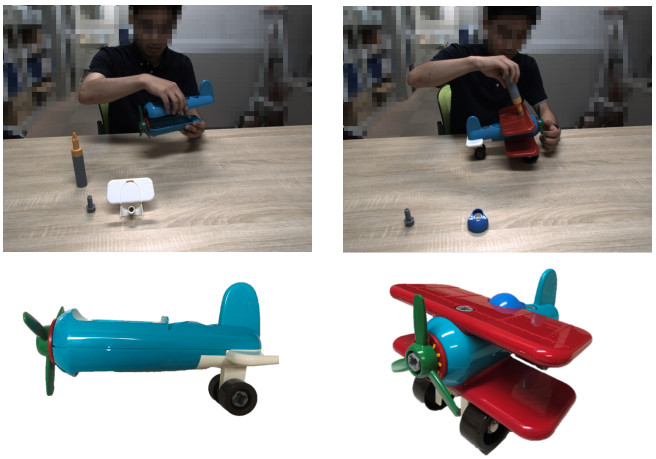
We employ several Hidden Markov Models [13] to recognize the motion corresponding to a specific object for the segmented motions (these HHMs are different from the one used in section II-D for differentiating hand states). An HMM is a stochastic process model composed of a time series data. The parameters of an HMM are the hidden state $\boldsymbol{S} = \{s_i\}$ $(i = 1, 2, ..., N)$, its transition state probability $\boldsymbol{A}$, and the probability $\boldsymbol{B} = \{b_{ij}\}$ that the hidden state s_i produces the observed state o_j. When the observation sequence $\boldsymbol{O} = \{o_k\}$ $(k = 1, 2, ..., T)$ is given, the probability $p(\boldsymbol{O}|\boldsymbol{\lambda})$ of the HMM $\boldsymbol{\lambda}$ generating the sequence $\boldsymbol{O}$ is computed using the Viterbi algorithm [14], this probability is called likelihood. In this case, where several HMMs are use for motion recognition, the resulting (recognized) motion is that of the HMM with highest likelihood $p(\boldsymbol{O}|\boldsymbol{\lambda})$. As explained above, an HMM is a sequence learning method, that is easy to implement for selecting and comparing multiple HMMs (it does not require a large dataset for training). For these reasons HMMs are used in this work.

We trained one HMM per pair of object and motion (42 in total). The number of hidden states was empirically determine to be $N = 5$ for all HMMs. The probability $\boldsymbol{B}$ and the state transition probability $\boldsymbol{A}$ are estimated using the featured data described on section II-B as the observed sequence $\boldsymbol{O}$ and the Baum-Welch algorithm [15]. We select the HMMs to be use for motion recognition from the combination of grasped objects (determine as described in section II-C) by both hands according to the action relationship define previously. For example, as shown in Fig. 7, when the assembly motion includes a bolt, the possible actions related to the bolt are selected as candidate motions, in this case "pick", "hold", and "move", then, their associated HMMs are use to compute their likelihood. If on the opposite hand, an object with screw holes such as body or wing is being grasped the HMMs associated to the "insert" and "screw" motions are also added as candidates. By training the HMMs corresponding to these motions per object, it has the effect of considering the motion variation due to size and shape of the manipulated object.

III. Experiments

In this section, we describe the experiment carried out to verify the validity of the proposed method. First, we explain about the data collection and the assembly tasks recorded.

(a) Assembly of composite parts (propeller, chassis, rear part)



(b) Assembly of the body part **(c)** Fully assembled airplane

Fig. 8: Assembly tasks of an airplane toy

TABLE I: Assembly actions list for the airplane task and main manipulated objects

Action	Manipulated objects	Description
pick	any	picking or grabbing an object
hold	any	holding or supporting an object
move	any	putting on the table or moving
insert	bolt & nut, chassis & wheel	insertion action
screw	bolt & nut, bolt/nut & screwdriver	screwing and tightening
place	body, wing, window(cockpit)	placing in the proper position of another part

Next, we describe about the evaluation criteria used. Finally, we discuss the obtained results.

A. Data Collection

To verify the validity of the proposed method, we use the task of assembling an airplane toy. This task includes motions using small parts such as bolts, nuts, etc., and also motions using a screwdriver to screw bolts, as shown in Table I. Due to working space constraints (lack of space) and the poor stability of the object detection YOLO when multiple objects are in the same image, we divided beforehand the assembly process of the airplane toy into three parts as shown in Fig. 8.

TABLE II: Average accuracy and standard deviation (in parenthesis) for all trials. "Case A" is when only the frames with the object correctly identified are used. "Case B" is when the object information is given from the Ground Truth data (all frames are used). "Case C" is when both the object and the segmentation information is given from the Ground Truth data (all frames are used).

	Object identification	Proposed method	Comparison method	Case A		Case B		Case C	
				Proposed method	Comparison method	Proposed method	Comparison method	Proposed method	Comparison method
Overall accuracy	0.616 (0.131)	0.410 (0.112)	0.373 (0.127)	0.664 (0.088)	0.591 (0.131)	0.729 (0.076)	0.665 (0.085)	0.823 (0.094)	0.760 (0.092)
Accuracy by object/motion type	0.502 (0.109)	0.265 (0.076)	0.197 (0.075)	0.514 (0.105)	0.446 (0.113)	0.530 (0.078)	0.418 (0.088)	0.686 (0.110)	0.570 (0.125)

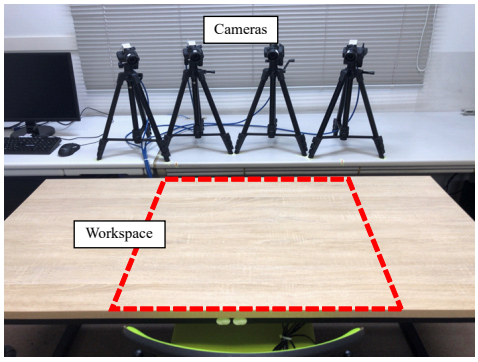


Fig. 9: Environment setup for experiments

The assembly motion data was obtained through the method explained in section II-B. The experimental setup used in this work is shown in Fig. 9. The four cameras were synchronized to record video at a 10 fps rate. To obtain the 3D position of the human's motion, the user had to carry out the assembly task inside the field of view of the four cameras, as illustrated in Fig. 9. The motion data needed to train the HMMs was obtained from two subjects. To decrease the number of frames lost due to failure in the pose estimation of OpenPose, the assembly task was further divided in a total of seven processes (only when recording the training data), and each process was recorded 10 times (five per subject).

For the test data set, we recorded the assembly motion of five subjects (different from the subjects recorded for the training data), two times per process per subject (30 in total). We segmented the motions of each hand and labeled them with their corresponding motion name to generate the Ground Truth of the test data.

To train the object detector YOLO with the airplane toy parts, we took 20 images (approx.) of each airplane toy part and tool (alone). We also used around 100 images from the ones taken to train the HMMs (where the objects are clearly visible), and labeled the object's name and bounding box by hand for YOLO to learn the objects used in this work.

B. Evaluation

To evaluate the proposed method, we compared the motion and object label of each recorded frame with the Ground Truth data and compute the rate of labels matching it. We computed the overall accuracy and the accuracy per type of motion for each assembly task trial.

Furthermore, we also evaluated the proposed method when we do not narrow down the candidate actions based on the action relationship of the manipulated objects. Namely, from the data of the motions listed in Table I, we trained six HMMs (one per motion) without distinguishing the manipulated object (we called this the "comparison method"). The motion segmentation procedure is the same, and then, the recognized motion is the one with highest likelihood among the six trained HMMs. As the proposed method heavily depends on the correctly identification of the manipulated object and segmentation of the data, we evaluated the proposed and the comparison method when the manipulated object was correctly identified only. Additionally, we verified the effect that the proposed segmentation method has over the recognition rate, by giving the identified object information from the Ground Truth data as an input. Finally, we also examined the motion recognition method performance, by giving the identified object and the segmentation points from the Ground Truth data as inputs.

C. Results and Discussion

Using the evaluation criteria described in section III-B, we evaluated each of the 30 test trials. The average and standard deviation are shown in Table II.

The accuracy of the manipulated object identification over all trials was 0.616. The performance of the manipulated object identification mainly depends on the object recognition (YOLO) accuracy, and the HMM that updates (or not) the object information when the object recognition fails (as described in section II-C). Therefore, we can conclude that the feature of the motion data used for this HMM was not enough to correctly identify the manipulated object. The overall accuracy of the proposed method was 0.410 and that of the comparison method was 0.373.

Isolating the effect of the manipulated object identification (i.e. considering only the frames where the manipulated object was correctly identified, refer as "Case A"), the proposed method accuracy was 0.664 and the comparison method accuracy was 0.591. From this results, we can see that the motion recognition accuracy increased twice its value, which means that the performance of the manipulated object identification greatly influences the motion recognition accuracy.

On the other hand, when the correct manipulated object is given and the segmentation is completed using hysteresis thresholding ("Case B"), the accuracy of the proposed and the comparison method were 0.729 and 0.665, respectively. This means, that the impact of the segmentation accuracy based on hysteresis thresholding is lower than that of the manipulated object identification accuracy. However, one restriction of the hysteresis thresholding based segmentation is that we need to determine empirically an appropriate threshold to improve its performance. As the speed of doing the assembly task changes depending on the type of task, the workspace, etc., it is hard to generalized. Therefore, a method to automatically determine this threshold or a feature that do not depends on speed is necessary to carry out a robust segmentation. When both the object identification and the segmentation information is given from the Ground Truth data ("Case C"), the accuracy of the proposed motion recognition method was 0.823 and that of the comparison method was 0.760.

Nevertheless, in any of the presented cases, the proposed method had a better performance that the comparison method. This means that narrowing down the candidate motions based on the action relationship between manipulated object and motion, contributes to the improvement of the motion recognition performance.

Finally, regarding the average accuracy per motion overall motion types, it can be seen that is significantly lower than the overall accuracy. This is partially due to the unbalanced duration of some motions, i.e. pick motions are very short in comparison with screw or insert. This means, the data used (feature) to compute the likelihood of the HMMs is few, increasing the probability of failing to correctly recognize the picking motion.

IV. Conclusions

In this work, we proposed a framework for the automatic motion segmentation and recognition of assembly tasks done by humans. First, we described how to obtain the human's pose and the assembly parts and tools positions through a sequence of images (video) using OpenPose and YOLO. From the obtained pose, we extract the joint's angles of the human's thumb, index and middle fingers as well as the wrist velocity and the width between fingers. We trained one HMM per pair of object-action based on the data of two subjects. We evaluated the proposed method using the motion data of five subjects assembling an airplane toy. The experimental results showed that the performance of the proposed method based on action relationships is better than when we only use motion data.

In the future, we would like to:

- improve the accuracy of the manipulated object identification method by introducing a pressure sensor on the human's finger tip or by extracting an appropriate motion feature;
- contemplate a method where the segmentation threshold is automatically determine or not needed;
- improve the recognition performance by using feature selection;
- use a different machine learning method for motion recognition (e.g. deep learning) and compare results;

among others. Then, we would like to go on and apply our method for the robot motion generation of assembly tasks.

ACKNOWLEDGMENT

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

[1] N. Yamanobe, W. Wan, I. G. Ramirez-Alpizar, D. Petit, T. Tsuji, S. Akizuki, M. Hashimoto, K. Nagata, and K. Harada, "A brief review of affordance in robotic manipulation research," *Advanced Robotics*, vol. 31, no. 19-20, pp. 1086–1101, 2017.

[2] R. Hanai and K. Harada, "A framework for systematic accumulation, sharing and reuse of task implementation knowledge," in *2016 IEEE/SICE International Symposium on System Integration (SII)*, Dec 2016, pp. 434–440.

[3] T. Mori, Y. Nejigane, M. Shimosaka, Y. Segawa, T. Harada, and T. Sato, "Online recognition and segmentation for time-series motion with hmm and conceptual relation of actions," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug 2005, pp. 3864–3870.

[4] E. E. Aksoy, M. Tamosiunaite, and F. Wörgötter, "Model-free incremental learning of the semantics of manipulation actions," *Robotics and Autonomous Systems*, vol. 71, pp. 118–133, 2015.

[5] A. Kubota, T. Iqbal, J. A. Shah, and L. D. Riek, "Activity recognition in manufacturing: The roles of motion capture and semg+inertial wearables in detecting fine vs. gross motion," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 6533–6539.

[6] A. Malaisé, P. Maurice, F. Colas, and S. Ivaldi, "Activity recognition for ergonomics assessment of industrial tasks with automatic feature selection," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1132–1139, April 2019.

[7] K. Fukuda, I. G. Ramirez-Alpizar, N. Yamanobe, D. Petit, K. Nagata, and K. Harada, "Recognition of assembly tasks based on the actions associated to the manipulated objects," in *2019 IEEE/SICE International Symposium on System Integration (SII)*, Jan 2019, pp. 193–198.

[8] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields," in *arXiv preprint arXiv:1812.08008*, 2018.

[9] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7291–7299.

[10] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1145–1153.

[11] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4724–4732.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 779–788.

[13] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[14] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[15] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.