Recognition of Assembly Tasks Based on the Actions Associated to the Manipulated Objects

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

***Abstract*— This paper proposes a complete framework to automatically recognize assembly manipulation motions performed by humans, for the purpose of generating and retrieving robot motions from a database. Using the concept of affordance, we can obtain the relationship between the manipulated object and its associated human actions to narrow down the possible actions that each manipulated object can afford. Based on this relationship we design motion templates containing a set of basic motions associated to the manipulated objects and stored them in the database. Recognition of motion data is done by matching it with the existing motion templates on the database using Hidden Markov Models (HMMs). We verify the validity of the proposed method using three different assembly tasks performed by two subjects, which include basic assembly motions such as insertion and bolt screwing.**

I. Introduction

In recent years, research towards the construction of a cloud database for assembly tasks has been developed with the aim at easing the motion generation of robotic tasks [1], as illustrated in Fig. 1. Hanai et al. [2] have constructed a framework for sharing and re-using teaching data for robot motions. In this framework, human motions are stored in a database and used as teaching data for robot motions. Therefore, it is very important that the data is stored in an efficient way to be easily retrieved and re-used. For this reason, it is necessary to divide into segments the motion data and assign a name tag (label) representative of each segmented motion. However, in most databases this process is manually done and therefore time consuming. Having this as a motivation, the goal of this work is to develop a framework able to automatically segment and recognize assembly motions done by humans.

Previous work on motion recognition and segmentation have focused mainly on daily tasks. Mori et al. proposed a method for the recognition of daily motions using a full body motion capture and Support Vector Machine (SVM) [3]. Based on these results, they proposed another method using time-series Action Probability to compute the likelihood of an action occurrence [4]. They use Hidden Marcov Models to analyzed the Action Probability to determine segmentation points. Aksoy et al. proposed a classification framework for

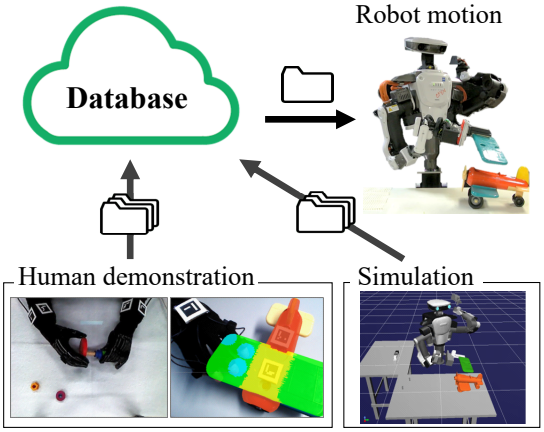


Fig. 1: Outline of a cloud database system for generating robot motions.

manipulation actions, where they use Semantic Event Chains (SECs) to represent the changes in the relationship between hand and manipulated objects [5]. However, the changes are based on the contact state between the object and hand, thus actions that do not have contact changes cannot be recognized.

On the other hand, Koppula et al. discussed the recognition of human actions and object affordances from RGB-D videos [6]. They formulated the problem as a Markov Random Field (MRF) and use SVM to learn the parameters of the MRF. Nevertheless this work also focused on whole body motions of daily activities (pouring water, opening a refrigerator, etc.).

None of the related work has discussed assembly tasks, since they require precise and highly skilled manipulations with fine motions that are difficult to detect and segment into basic motions. In this work, we use the concept of object affordances to narrow down the possible actions (candidate actions) that an object can afford. Object affordances [7] are any possible action that a human or animal can execute on an object or the environment. Using this concept, we can define the relationship between the manipulated object and its associated human actions. Based on this relationship we design motion templates containing a set of basic motions associated to the manipulated objects and stored them in a database. It must be noted that as one object can afford different actions and can be used in the different assembly tasks, it is less time consuming defining its action relationship than manually labeling all the assembly tasks associated to it. Next, motion recognition and segmentation are executed

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

[2]Manipulation Research Group, Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba, 305-8560, Japan {n-yamanobe, k-nagata}@aist.go.jp

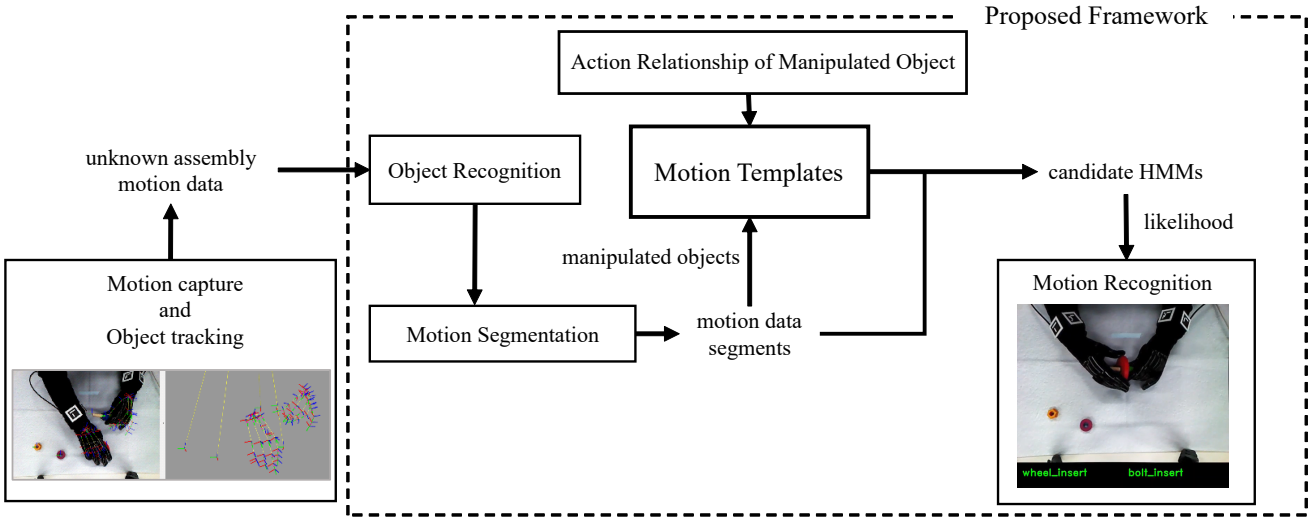


Fig. 2: Outline of the proposed framework for motion recognition. The relationship between objects and possible actions is stored together with its associated Hidden Markov Models (HMMs) on a motion template. To recognize an unknown sequence of motions, we first recognize the objects being manipulated and segment the sequence. A series of candidate HMMs are retrieved based on the action relationship of manipulated object in each segment. Then, the segmented motion data is used to compute its likelihood of being generated by each of its candidate HMM. Finally, the motion associated to the HMM with the highest likelihood is considered to be the unknown motion.

by matching the existing motion templates on the database and the captured data. We verify the validity of the proposed method using a wooden-made assembly kit, data-gloves, AR Markers, and an RGB-D camera, to obtain the motion data of two subjects performing three different assembly tasks.

This paper is organized as follows: in section II the proposed framework for motion segmentation and recognition of assembly tasks is presented. Then, in section III we describe the motion capture system for acquiring motion data and show the obtained results for assembly motions. Finally, in section IV we give the conclusion of this work and discuss future work.

II. Motion Recognition Framework

In this section, the proposed framework (Fig. 2) for motion recognition of assembly tasks is described. First, we introduce the relationship between the manipulated object and its possible actions for assembly tasks. Next, we explain how to recognize the motion data by using the previously introduced relationship. Finally, we introduce a motion template in which motion data is associated to an object and its action relationship.

A. Action Relationships of Manipulated Objects

As mentioned in section I, affordance refers to the possibility of an action on an object or the environment [7]. In this work, we exploit the concept of affordance to obtain the relationship between the manipulated object and the human actions afforded by it.We define as "primary object" to the object that is being manipulated (in motion), and we called "secondary object" to the object that is mainly supporting (holding) the action of the primary object. An example of the relationship between objects and actions is shown in Fig. 3,

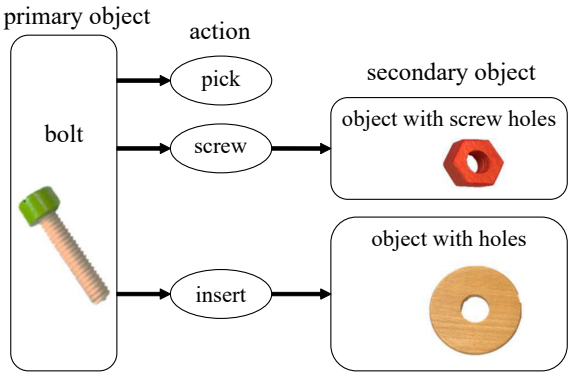


Fig. 3: Action relationship of a bolt.

where the relationship between the possible actions of a bolt and two secondary objects is illustrated.

If the manipulated objects are known, then, knowing the possible actions related to it is straightforward from the obtained action relationship. Therefore, the matching search on the database for the observed motion can be narrowed down to a finite number of possibilities, improving the efficiency of the recognition process. It should be pointed out that as one object can afford different actions and can be used in different assembly tasks, it is less time consuming defining its action relationship than manually labeling all the assembly tasks associated to it.

B. Motion Recognition

At first, a sequence of assembly motions is segmented at each point in which a manipulated object(s) change occurs. To determine this change, we define a manipulation threshold to detect which object is being manipulated by which hand, this will be explained in detailed in section

III-B. Like this, if the manipulated objects are identified during the entire assembly process, the observed motion can be segmented into basic motions. Based on the relationship between manipulated object and actions, a finite number of actions from the database are selected as candidate actions. Then, we employ Hidden Marcov Models (HMMs) [8] to recognize the segmented motions.

An HMM is a stochastic process model of a sequence of observed states. This HMM can be used to compute the probability $p(\boldsymbol{O}|\boldsymbol{\lambda})$ of an unknown sequence $\boldsymbol{O}$ of being generated by the same HMM, i.e. the unknown sequence is similar to the sequence used to construct the HMM.

In this work, an HMM is constructed to model the finger's joint-angles of each hand for each pair of object-action. Each HMM is represented by the following parameters:

1. Set of States $\boldsymbol{S} = \{s_i\}$ $(i = 1, 2, 3, ..., N)$,
2. Observation Sequence $\boldsymbol{O} = \{o_k\}$ $(k = 1, 2, ..., \infty)$,
3. State transition probability $\boldsymbol{A} = \{a_{ij}\}$, a_{ij} is the probability of going from state s_i to state s_j,
4. Observation symbol probability $\boldsymbol{B} = \{b_{ij}\}$, b_{ij} is the probability of state s_i generating the observation o_j,
5. Initial state distribution $\boldsymbol{\pi} = \{\pi_{s_i}\}$,

where the number of observed states is set to be $N = 3$, that correspond to the hand states of: open, close and in between open and close.

The parameters listed above are estimated using the Baum-Welch algorithm [9] and a set of motion data recorded using the experimental system described in section III. Each trained HMM $\boldsymbol{\lambda}$ is stored together with its associated action relationship of the manipulated object and motion information (a detailed explanation is given in section II-C).

The motion recognition is then carried out by computing the probability $p(\boldsymbol{O}|\boldsymbol{\lambda})$ of the candidate HMM $\boldsymbol{\lambda}$ having generated the observed sequence (unknown motion). This probability (also called likelihood) is computed for each of the candidate actions using the Viterbi algorithm [10]. Therefore, the observed motion is determined to be the same as that of the associated motion to the HMM with the highest likelihood, for each hand independently of the other.

C. Motion Template

As mentioned in section II-B, each trained HMM with its associated action relationship of the manipulated object and motion information are stored in a database. For this purpose, we define a storage format called "motion template", where besides storing motion data and its associated action relationship of the manipulated object and trained HMM, we also stored the manipulation threshold. Fig. 4 shows the proposed motion template for a bolt.

It should be noted that the proposed motion template can be applied to other type of manipulation tasks for storage, sharing and/or recognition purposes.

III. Experiments

In this section, we describe the motion capture system used in this work and show the obtained results for recognizing assembly motions. First, we show the assembly tasks used in this work. Next, we describe the motion capture system employed for acquiring the assembly motion data. Finally, we show the experimental results for 3 different assembly tasks.

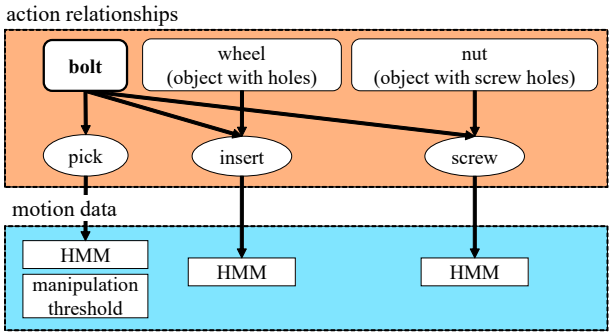


Fig. 4: Motion template of a bolt.

Fig. 5: F1 car model used in this work.

A. Assembly Task

To verify the validity of the proposed framework, we use the commercially available wooden kit: SEVI® construction kit F1 car. This assembly kit can be used to construct different models and it also includes a screwdriver and a wrench. In this work we will only use the F1 car model shown in Fig. 5. According to the relationship introduced in section II-A, the possible actions for each of the objects composing the F1 car are listed in Table I. An object composed by more than two single objects is called "composite object", Fig. 6 shows the composite objects of the F1 car model.

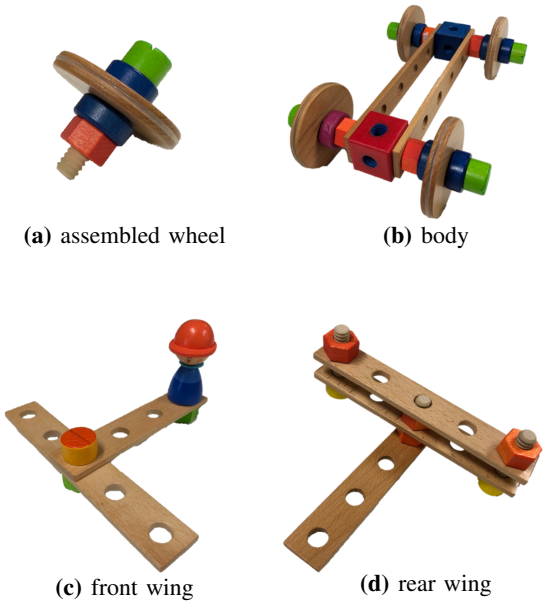
B. Motion Capture System

To acquire the motion data of the assembly task described in the previous section, we use a pair of data gloves (CyberGlove III [11]), an RGB-D (Orbbec Astra S [12]), and AR markers, as shown in Fig. 7. The sequences of 22 joints' angles (per hand) and hand/object positions compose what we called motion data and are stored in the motion template. The interphalangeal and metacarpophalangeal joints of the thumb, index, and middle fingers, and thumb abduction (total 9 sequences) are selected for training HMM. We employed the joints' kinematic model of Wang et al. [13]. These data are obtained from the data gloves with a sample frequency of 10 Hz.

For determining which object is being manipulated by which hand and apply the action relationship in the motion template to recognize assembly motions, we need to have

TABLE I: Assembly motion list for the F1 car model

Primary object	Type of motion	Secondary object
bolt	pick	
	insert	board, wheel, wheelcover
	screw	nut, cube
nut	pick	
	screw	bolt
cube (with screw holes)	pick	
	screw	bolt
board	pick	
	pile up	board, cube
wheel	pick	
	insert	bolt
wheelcover	pick	
	insert	bolt
screwdriver	pick	
	screw	bolt
wrench	pick	
	wrench	nut



(a) assembled wheel **(b)** body

(c) front wing **(d)** rear wing

Fig. 6: Composite objects of the F1 car shown in Fig. 5.

knowledge of the objects being manipulated. For this particular case of the wooden toy and for the sake of simplicity, we use a particle filter to track specific colors to detect the position of the objects. Also, AR markers are employed to detect the position of the composite objects and the position of the hands, that together with the data gloves are used to compute the fingertips positions. Using the object and the fingertips positions, we define a manipulation threshold to determine if an object is being manipulated by a hand or not. If the position of the object (relative to the fingertips position) exceeds the defined threshold for the thumb, index

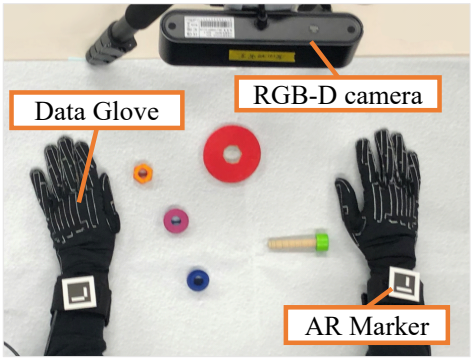


Fig. 7: Overview of the motion capture system.

and middle fingers, it is determined that the object is not being manipulated. In case the positions of multiple objects are under the defined threshold, the object with the closest distance to the middle finger is selected as the manipulated object. As the grasping configuration is different for each object, the manipulation threshold was experimentally defined for each object (about 5 cm). This manipulation threshold was determined based on the experimental data used for training the HMMs and it is stored in the motion template.

C. Results and Discussion

As mentioned in section II-B, to estimate the parameters of each HMM, a set of training data is necessary. Using the motion capture system described in the previous section, we gathered a training data set composed of the single object motions listed in Table I. Each motion was done when holding the primary object with each hand, and repeated 15 times per hand.

To verify the validity of the proposed framework, we recorded a test data set composed of both single object and composite object motions of 3 different assembly tasks:

1. Wheel assembly (single object)
2. Partial body assembly (composite object)
3. F1 car final assembly (composite object)

Each assembly task was recorded 5 times by two different subjects (subject "A" being the same person that recorded the training data).

To evaluate the test data set, we prepared a set of ground truth data by manually segmenting and identifying the assembly motions at each time sample (10 Hz). We define the success rate as the number of correctly recognized frames divided by the number of recorded frames per assembly task. Thus, if the experimental segmentation differs from the ground truth, the recognition result is taken as incorrect. Table II shows the average of the success rate obtained for each assembly task over the recorded test data set (5 trials). In this table, "object recognition" percentage refers to the correctly recognize objects when manipulated by any of the hands. The "motion recognition" percentage represents the correctly recognized motions when the manipulated object was correctly recognized. Here and after we will discuss only the motion recognition results since the object recognition

TABLE II: Average success rate [%]

	Assembly task	Object recognition	Motion recognition
Subject A	wheel	81.6	60.5
	body	45.9	64.0
	F1 car	57.4	44.5
	average	61.6	56.4
Subject B	wheel	69.8	52.8
	body	46.7	81.0
	F1 car	67.0	52.8
	average	61.1	62.2
Average subjects A and B	wheel	75.7	56.7
	body	46.3	72.5
	F1 car	62.2	48.7
	average	61.4	59.3

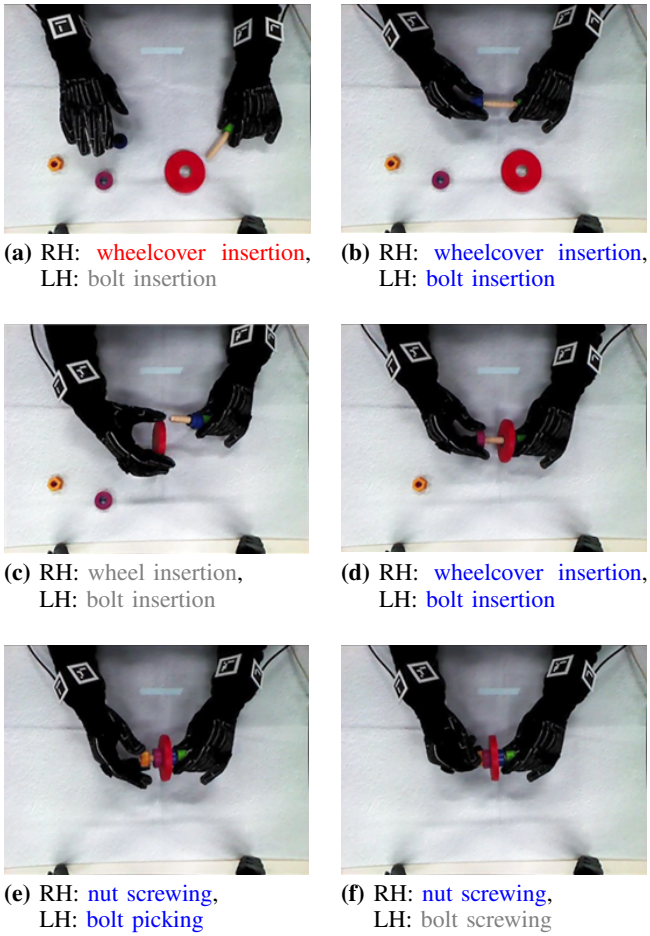
TABLE III: Average success rate per motion

Type of motion	Success rate (%)	Frequency (%)
pick	22.8	49.5
screw	28.8	24.0
pile up	41.7	6.0
insert	46.1	9.1
no motion	48.3	11.4

depends mainly on the vision system and it is out of the scope of this work. It should be noted that although it would have been easier to assume that the manipulated object was known, everything was experimentally obtained. Fig. 8 shows snapshots of the recorded task when assembling the wheel, the subcaptions show the recognized motions. In red are written the incorrectly recognized objects and motions, in gray the correctly recognized objects but incorrectly recognized motions, and in blue the correctly recognized objects and motions.

As mentioned before, the success rate includes not only the performance of the trained HMMs but also that of the segmentation process, which relies mainly on the recognition of the manipulated object. For example in Fig. 8(a), the left hand is not holding any object but the system incorrectly detects the wheelcover as being in hold by the left hand. Also, as the segmentation is based on a manipulated object change (i.e. the manipulated object changes), it was difficult for the system to break the pieces of motion starting with picking (e.g. pick and insert, which for the HMMs training was considered as two different motions). In those cases where the segmentation failed to separate two different motions, the recognition result was that of the motion that took a longer execution time. Therefore, the picking motion recognition success rate was the lowest among all the motions, as shown in Table III. In Figs. 8(a), (c) it can be seen that when the object is being pick the system recognizes the motion as insertion.

To isolate the influence of this segmentation problem in the recognition rate, we prepared a new set of ground truth data



(a) RH: wheelcover insertion, LH: bolt insertion
(b) RH: wheelcover insertion, LH: bolt insertion
(c) RH: wheel insertion, LH: bolt insertion
(d) RH: wheelcover insertion, LH: bolt insertion
(e) RH: nut screwing, LH: bolt picking
(f) RH: nut screwing, LH: bolt screwing

Fig. 8: Snapshots of the wheel assembly task by time execution order. The recognized motions are written at each subcaption, where LH means left hand, and RH right hand. Written in red are incorrectly recognized objects and motions, in gray are correctly recognized objects but incorrectly recognized motions, and in blue are correctly recognized objects and motions.

where we consider a pick motion followed by an insertion motion as a single segment of insertion, and the same for a pick motion followed by a screw motion (a single screw segment). The average success rates with the new ground truth are shown in table IV. It can be seen that overall, the recognition improved around 3.7%.

Regarding the versatility of the system, from table IV it can be seen that the difference in the success rate between subject A (same person that recorded the training data set) and B, across all the evaluated assembly tasks is only 1.7%, and it is higher for subject B. In the particular case of the body assembly task, subject B has an almost 20% higher success rate than subject A. These results demonstrate the versatility of the proposed framework. As shown in table V, even though the assembly task average execution time for subject B is considerably shorter (around 34%) than subject A for the F1 car assembly task, the difference in success rate is only 1.3%. This means that the execution time of the assembly task does not significantly influence the recognition

TABLE IV: Average success rate when considering successive motions [%]

	Assembly task	Motion recognition
Subject A	wheel	72.6
	body	62.1
	F1 car	51.6
	average	62.1
Subject B	wheel	57.8
	body	80.9
	F1 car	52.9
	average	63.9
Average subjects A and B	wheel	65.2
	body	71.5
	F1 car	52.3
	average	63.0

TABLE V: Execution time comparison [s]

assembly task	Subject A	Subject B
wheel	$41.5 \pm 12.2\%$	$36.1 \pm 4.5\%$
body	$66.3 \pm 13.7\%$	$58.3 \pm 19.0\%$
F1 car	$53.1 \pm 13.9\%$	$34.9 \pm 20.9\%$

results. Also, although assembly tasks such as picking and screwing could change considerably from subject to subject due to personal habits, it can be seen that the proposed framework is able to recognize more than half of the tested data.

IV. Conclusions

This paper proposed a complete framework to automatically recognize assembly manipulation motions performed by humans to generate new robotic assembly tasks. The main results of this paper are summarized as follows:

1. We introduced a relationship between manipulated object and possible actions for assembly tasks. Using this relationship, we are able to narrow down the search for candidate motions.
2. We proposed a motion template for storing motion data on a database. In this template, the relationship of manipulated object to each action together with a manipulation threshold use for segmentation are stored to efficient the search process for data.
3. We recognized assembly motions using Hidden Markov Models based only on the data of three fingers' joint-angles of each hand.
4. We showed experimental results of three different assembly tasks executed by two different subjects to verify the validity of the proposed framework.

The results showed that the proposed framework is versatile and robust to changes in execution time and type of assembly tasks. In the future, we would like to improve the segmentation process and consider other motions such as holding, in order to improve the overall recognition success rate. Also, we used object's colors and AR markers to track the objects. However, the assembly objects are often metalic and attaching markers to the objects may interfere with the natural process of assembly. Thus, in the future we would like to use an object detector to track the objects. Object detectors (e.g. YOLO [14]) are expected to be useful for applying our framework to all kinds of objects and 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. [Online]. Available: https://doi.org/10.1080/01691864.2017.1394912

[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, M. Shimosaka, T. Harada, and T. Sato, "Recognition of actions in daily life and its performance adjustment based on support vector learning," *International Journal of Humanoid Robotics*, vol. 01, no. 04, pp. 565–583, 2004. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0219843604000368

[4] 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.

[5] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889014002450

[6] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 951–970, 2013.

[7] J. J. Gibson, *The theory of affordances, in Perceiving, Acting, and Knowing. Towards an Ecological Psychology.* Hoboken, NJ: John Wiley & Sons Inc., 1977, no. eds Shaw R., Bransford J.

[8] 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.

[9] 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. [Online]. Available: http://www.jstor.org/stable/2238772

[10] 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.

[11] Cyber Glove Systems Inc., "Cyber Glove III," http://www.cyberglovesystems.com/, (accessed on 2/12/2018).

[12] Orbbec 3D Tech. Intl 社, "Orbbec Astra S," https://orbbec3d.com, (accessed on 2/12/2018).

[13] Y. Wang and M. Neff, "Data-driven glove calibration for hand motion capture," in *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '13. New York, NY, USA: ACM, 2013, pp. 15–24. [Online]. Available: http://doi.acm.org/10.1145/2485895.2485901

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