2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Wiping 3D-objects using Deep Learning Model based on Image/Force/Joint Information

Namiko Saito[1], Danyang Wang[1], Tetsuya Ogata[2], Hiroki Mori[3] and Shigeki Sugano[1]

***Abstract*— We propose a deep learning model for a robot to wipe 3D-objects. Wiping of 3D-objects requires recognizing the shapes of objects and planning the motor angle adjustments for tracing the objects. Unlike previous research, our learning model does not require pre-designed computational models of target objects. The robot is able to wipe the objects to be placed by using image, force, and arm joint information. We evaluate the generalization ability of the model by confirming that the robot handles untrained cube and bowl shaped-objects. We also find that it is necessary to use both image and force information to recognize the shape of and wipe 3D objects consistently by comparing changes in the input sensor data to the model. To our knowledge, this is the first work enabling a robot to use learning sensorimotor information alone to trace various unknown 3D-shape.**

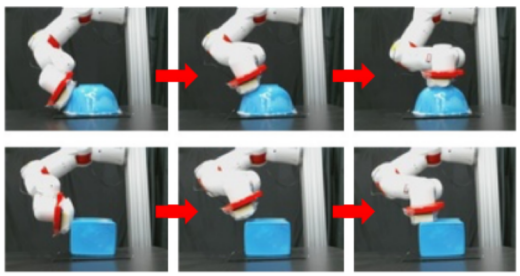

Fig. 1. The robot needs to wipe continuously if the target has a round shape, but needs to change the angle by 90 degrees along the surface if the target has a cuboid shape.

I. INTRODUCTION

Robots capable of working alongside humans and performing daily tasks automatically is becoming an increasingly important focus of research in the field of robotics [1]. In a study by Cakmak et al. [2], household tasks were classified and it was shown that cleaning tasks accounted for 49.8% of all chore tasks. Here, we focus on the task of wiping objects, which is one of the most basic cleaning tasks.

There are many objects that need to be wiped in our daily lives, such as tableware, light bulbs, shelves, bathtubs, and statues. Features such as shape, deformations, and hardness differ among various objects. In addition, there is a nearly infinite number of possible objects, and it is difficult to trace objects of arbitrary shape. Here, we present as a first step of research into wiping various types of furniture.

Wiping is a task that consists of interacting with objects. Robots need to consider "the wiping method" depending on the shape of the target object. For example, as shown in Fig. 1, if the target has a round shape, it is better to wipe continuously. However, if the target has a cuboid shape, then it is necessary to plan changes in angle of 90 degrees to wipe according to the surface. In addition, the directions of the wiping motions need to be decided according to the inclination of the surfaces. Therefore, a robot needs to recognize the shape, plan an appropriate wiping method, and adjust its movements accordingly.

One of the most difficult problems faced by current robots is adaptability to target objects. This is because most robots are designed to repeat the same specific motions for certain specific objects. When it becomes necessary to handle a new object, the control system or the computational model needs to be redesigned. We therefore propose a model which can adjust to unknown objects by using only the sensor information available at the time.

We will realize robot wiping 3D-objects with our deep neural network (DNN) model. The model comprises two modules: a convolutional autoencoder (CAE) [3] and a multiple timescales recurrent neural network (MTRNN) [4]. The CAE compresses the raw image and generalizes the untrained images while the MTRNN estimates the shape of the object and generates the appropriate wipe actions based on real-time sensorimotor feedback. The model is tested on the Torobo robot arm developed by Tokyo Robotics [5]. For evaluation, we use untrained 3D objects and confirm whether the robot can wipe them in order to show the generalization ability of the DNN model. We also analyze the role of image and force information for wiping by comparing the success rate as the combination of input sensorimotor data is varied.

Our contributions are as follows.

- The robot enables planning that considers "the wiping method" depending on the shape of a 3D object, and wipes the object based on image, force, and joint information.
- The model does not require a pre-designed model of the target objects, and the robot can handle unknown objects on the spot.

*This research was partially supported by the JSPS Grant-in-Aid for Scientific Research (A) No. 19H01130, and Research Institute for Science and Engineering of Waseda University.

[1]Namiko Saito, Danyang Wang and Shigeki Sugano are with Department of Modern Mechanical Engineering, Waseda University, Tokyo, Japan n_saito@sugano.mech.waseda.ac.jp, d_wang@suganos.mech.waseda.ac.jp, Sugano@waseda.jp

[2] Tetsuya Ogata is with the Department of Intermedia Art and Science, Waseda University, Tokyo, Japan, and National Institute of Advanced Science and Technology, Tokyo, Japan ogata@waseda.jp

[3] Hiroki Mori is with the Future Robotics Organization, Waseda University, Tokyo, Japan mori@idr.ias.sci.waseda.ac.jp

II. Related Works

Several studies have investigated the planning of wiping trajectories by robots. Leidner et al. [6] developed a system for a robot to wipe a desk according to a computational model. The wiping motions were separated into three categories according to the dirt type (absorbing, collecting, and skimming), and the system planned the Cartesian trajectories. Cauli et al. [7] and Martínez et al [8] controlled a robot to clean the dirt on a specific table. Cauli et al. used an overhead camera and the robot's eye camera, and Martínez et al used an RGBD camera to find the locations of dirt. Their system planned the wiping trajectory of the robots' arm. However, that research used only image information and wiped a specific flat target. In addition, they were unable to wipe 3D objects.

There have also been several studies which used force feedback to wipe targets. Sato et al. [9] presented a robot system for cleaning a whiteboard. The robot was trained by imitation learning using hybrid position/force control and executed wiping of the whiteboard. Gams et al. [10] also used force feedback to learn dynamic motions, and ensured the robot maintained contact and applied the desired force to the target. Gams et al. enabled the robot to wipe not only a flat table but also a raised surface. However, the robot could handle only simple objects that could be wiped continuously with one motion without switchbacks, and was not able to recognize and conduct the suitable "wiping method," or adapt to more complex objects.

Some studies have tackled wiping 3D objects. Hess et al. [11] proposed an algorithm to cover a region of a 3D surface using a Kinect sensor to obtain a point cloud and generate an explicit model of the surface. That system calculated the optimal trajectory to cover the scanned surface. Nagata et al. [12] presented a position/force controller implementation for a polishing robot that used a CAD model of the surface. However, those studies required some procedures before new objects could be wiped because they required 3D surface scanning or a 3D model of the object in advance.

In our research, we propose a learning model for generating joint angles for 3D object wiping motions according to the shape of the object without computational models, using both force and image feedback from touching the object.

III. Method

We use the direct teaching method to control the robot for wiping and obtaining training data. In the direct teaching method, the robot arm is moved directly by the experimenter. We then collect the sensorimotor data as training data while replaying the movement of the direct teaching. If we had used the data obtained by direct teaching for training, the model would mix the feedback from the experimenter and from the target object because the force data includes the force produced by the experimenter and the image data would also include the arm of the experimenter. We therefore collect the training data during the replay.

The collected data is then used to train the two deep learning modules. The model learns to recognize the shapes of objects and generate wiping motions by the robot itself. Finally, we evaluate the wipe-motion generalization ability of the DNN model using untrained objects.

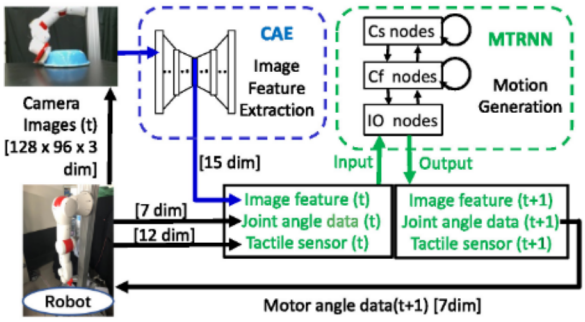

Fig. 2. Overview of the DNN model consisting of a CAE that extracts image features and MTRNN that generates next step motion.

TABLE I

Setting of MTRNN nodes

Nodes	Time constant	Number of nodes
Cs	50	13
Cf	5	160
IO	2	34 (Joint(7)+Tactile(12)+Image(15))

TABLE II

Setting of feedback rate α

Situation	Data	Value of α
Training	All sensorimotor data	0.9
Evaluation Experiments	Joint angle	1.0
	Image feature & tactile sensor data	0.2

IV. Deep Learning Model

Figure 2 shows an overview of the model, which consists of two components: a CAE and MTRNN. This model is taken from a previous study [13].

A. Image feature extraction by CAE

CAE has an hourglass-like structure that learns to make the output data the same as the input data. In this way, the raw image can be compressed, and the features of the image can be extracted from the intermediate layer while using a lower number of dimensions.

The construction of the CAE is shown in Fig. 3. We used the ReLU function for the activation function. Although the raw camera images initially have 36,864 dimensions [128 (width) $\times$ 96 (height) $\times$ 3 (channels)], the data is compressed into 15-dimensional image features. The picture on the right in Fig. 3 is the output image, which is reconstructed to resemble the input picture on the left in Fig. 3. The model performs well at compressing the essential information into image features for subsequent restoration.

B. Motion generation by MTRNN

MTRNN is a type of recurrent neural network that can predict and generate the next step when given the current state. Unlike a conventional RNN, the MTRNN comprises

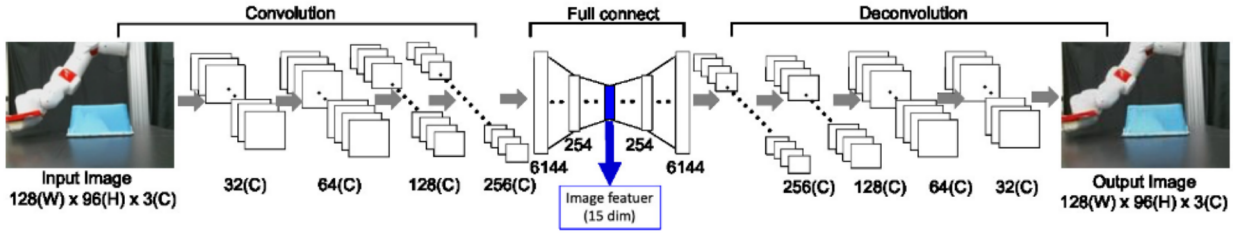

Fig. 3. Construction of CAE. The network compresses large dimensional image data into 15 dimensional image features.

three types of nodes with different time constants, which are slow context (Cs) nodes, fast context (Cf) nodes, and input/output (IO) nodes. Because of their large time constant, Cs nodes are expected to learn the data sequence, whereas Cf nodes with their small time constant are expected to learn the detailed motion primitives. This enables learning of the long-term dynamics of time-series data.

We set the time constants and the number of each node as shown in table I. We tried using a number of Cs nodes in the range of 10 to 20, time constant of Cs nodes in the range of 30 to 60 steps, and Cf node in the range of 100 to 200 steps in steps of 10. We adopted the best combination for the best training. If these numbers are too small, complex information cannot be learned, and if they are too large, the model is overtrained. The time constant of Cf did not have much effect even when it was changed to around 5.

In the forward calculation of the MTRNN, the output values are calculated as follows. First, the internal value u_i of the neuron i at step t is calculated as

$$u_i(t) = \left(1 - \frac{1}{\tau_i}\right) u_i(t-1) + \frac{1}{\tau_i}\left[\sum_{j \in N} w_{ij} x_j(t)\right], \quad (1)$$

where N is the number of neurons connected to neuron i, τ_i is the time constant of neuron i, w_{ij} is the weight value from neuron j to neuron i, and $x_j(t)$ is the input value of neuron i from neuron j. The output value is then calculated using the tangent hyperbolic function as the activation function as

$$y_i(t) = tanh\left(u_i(t)\right). \quad (2)$$

The value of $y_i(t)$ is used as the next input value as

$$x_i(t+1) = \begin{cases} \alpha \times y_i(t) + (1-\alpha) \times T_i(t+1) & i \in IO \\ y_i(t) & otherwise \end{cases}, \quad (3)$$

where $T_i(t)$ is the input datum i, which is training data during model training or real-time data during evaluation experiments. If neuron i is an IO node, the input value $x_i(t)$ is calculated by multiplying the output of the preceding step $y_i(t-1)$ and the datum $T_i(t)$ by the feedback rate α $(0 \leq \alpha \leq 1)$. We can adjust the input data by means of the feedback rate α.

We set the value of α as shown in table II. During training, the model can be trained efficiently by setting α to 0.9, that is, combining the predicted data 90 % and training data 10 %. We tried setting α to 0.8, 0.9, and 1.0, and finally adopted 0.9 which gave the best training. When we evaluate the model, the next position of the motor angle needs to be decided based on the previous position. Thus, we set the value to 1.0 to generate motor data according to the predicted data. In contrast, we use the value to 0.2 for image feature and tactile sensor data so that the model mixes real time data with the previous prediction in order to make it possible to adapt to each individual circumstances.

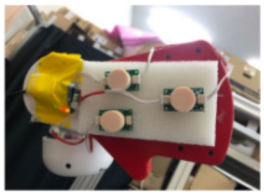

Fig. 4. The robot hand which is attached 3 tactile sensors.

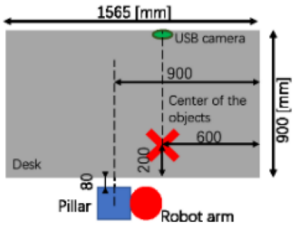

Fig. 5. Table setting.

In the backward calculation, we use the back propagation through time (BPTT) algorithm [14] to minimize the training error given by Eq. 4, and update the weight using Eq. 5

$$E = \sum_i \sum_{i \in IO} \left(y_i(t-1) - T_i(t)\right)^2 \quad (4)$$

$$w_{ij}^{n+1} = w_{ij}^n - \eta \frac{\partial E}{\partial w_{ij}^n} \quad (5)$$

where $\eta (= 0.0001)$ is the learning rate and n is the number of iterations.

V. Experimental Setup

A. System design

We use a Torobo arm which has 7 degrees of freedom and tactile sensors [15] which contain 4 sensing points per module. We attach 3 modules to the robot hand as shown in Fig. 4. We then cover the hand with thin cloth tape not to damage or break the sensors. In addition, although the sensors has a cylindrical shape, the surface of the hand can be flattened by applying a tape so that the robot can wipe objects smoothly. Further, the detection range of contact can be expanded. We put the tape gently and cover over the sensors so that to make the sensor value zero when the hand is not touching anything. Because the tape is thin enough, the force on the hand is applied directly to the sensor and the tape does not effect the recorded tactile sensor data.

TABLE III

SIZE OF OBJECTS

	objects	bottom width	upper width	height
Training	Big cube	268 [mm]	217 [mm]	113 [mm]
	Small cube	165	165	130
	Big bowl	300	150	94
	Small bowl	218	91	99
Evaluation experiments	Untrained 1	270	250	125
	Untrained 2	202	-	142
	Untrained 3	270	125	105
	Untrained 4	268	150	101

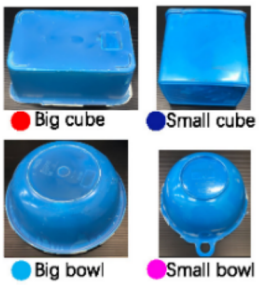

Fig. 6. Objects for training

Fig. 7. Objects for evaluation experiments

We put a desk in front of the robot and place a USB camera and objects as shown in Fig. 5. We place the objects so that the center is always at the same position, and arrange the largest surface parallel to the USB camera.

We record robot arm joint angle data (7 dim) and image features from the camera (15 dim) every 0.2 [sec]. We record tactile data (3 sensors $\times$ 4 dim) every 0.05 [sec] and average the sensor values over 4 steps and sample every 4 steps. The sampling frequency of all the sensorimotor data is then 5 [Hz]. The length of all wiping motions is 21.6 [sec], and thus they are 108 steps in every motion. The values of all sensorimotor data are normalized to [-0.9, 0.9] before input to the model.

B. Object used

Figure 6 shows the 4 objects used for training. We prepared 2 sizes of cubes and bowls. Figure 7 shows the 4 objects used for the evaluation experiments. We prepared different sizes and shapes of objects. The sizes of the objects are shown in Table III. In order not to distinguish the characteristics of the objects by their color, we painted all the objects the same shade of blue.

C. Task design

The robot starts every motion from a specific home position. The robot then wipes by moving from the bottom to the top of the left side of the object. Next, it wipes the upper side from left to right, and then from the top to the bottom of the right side.

We define the "wipe rule" as "when more than 0.1 [N] force is applied to the hand, it is regarded as wiping." This is checked by the tactile sensors. We determine the success of wiping as a criterion of whether more than two thirds of each side meets the wipe rule. We check if the robot recognize the method of wipe and conduct it considering the necessity of switchbacks, thus we do not care the amount of the force to be applied.

TABLE IV

RESULT OF EXPERIMENT 1

object	Left side	Upper side	Right side
Untrained object 1	100%	100%	100%
2	100	0	100
3	100	100	100
4	100	100	100

D. Training setup

The training data consists of a total of 32 (4 objects $\times$ 8 times) sets of time-series data.

We train the CNN module for 1000 iterations and the MTRNN modules for 20,000 iterations. In the final iteration, the training loss calculated by eq. 4 completely converged to a small value.

E. Evaluation experiments

1) Experiment 1: We control the robot such that it wipes the 4 untrained objects to check the generalization ability of the DNN model. We perform the test 3 times each. We also conduct principle component analysis (PCA) on the internal state of the MTRNN to evaluate if the robot can recognize the shape of the untrained objects. We perform the analysis at the 20th step of the Cs nodes because the 20th step is the time the robot has just started to touch the objects, and the distribution of the internal state of Cs also expands according to the way the wipe is performed.

2) Experiment 2: We check the contribution of image information and force information by changing the input data of MTRNN according to the following combinations. We compare the success rate of wiping with 4 training objects. We perform the test 6 times each.

- Image feature + Tactile sensor data + Joint angle (Proposal)
- Image feature + Joint angle
- Tactile sensor data + Joint angle

VI. RESULTS AND DISCUSSION

A. Experiment 1

Figure 8 shows the PCA results for the internal state at step 20 of the Cs nodes. We plotted the state of the training data as circles. The plots are colored according to the shape of the object being wiped. It can be seen that the different wiping methods are separated and similar methods are clustered. This shows that the DNN model could self-organize the shape of the objects as well as the wiping method.

We also plotted untrained objects in Fig. 8 as stars. The correspondence between plot colors and object shapes is as shown in Fig. 7. As shown in Fig. 8, the results for untrained

TABLE V

RESULTS OF EXPERIMENT 2, SUCCESS RATE OF EACH INPUT

Input	Big cube			Small cube			Big bowl			Small bowl		
	Left side	Upper side	Right side	L	U	R	L	U	R	L	U	R
Image + Tactile + Joint	100%	100%	100%	50%	83.3%	100%	100%	100%	50%	100%	100%	100%
Image + Joint	50	16.7	0	50	33.3	33.3	100	33.3	16.7	100	33.3	0
Tactile + Joint	0	0	0	0	0	0	100	16.7	100	100	16.7	0

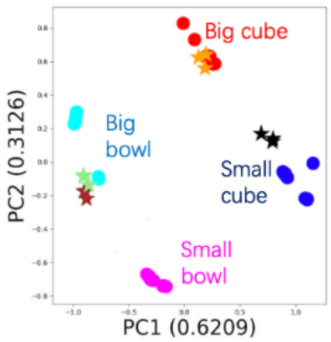

Fig. 8. Principal component analysis on the internal state of Cs nodes.

object 1 are near the region of trained big cube. Therefore, we assume that the robot predicted this object to be similar to a big cube. In the same way, we can assume that the robot predicted untrained object 2 as similar to small cubes, and the untrained object 3 and 4 as similar to big cubes.

Figure 9 shows how the robot generated wipe motions for untrained objects 1 and 4. When wiping untrained object 1, the robot switched the wiping direction according to the surface. The robot adjusted the wiping angle as the way to wipe cubes. When wiping untrained object 4, the robot used a continuous wipe as the way to wipe bowls.

We show the trajectory of motor angles during wiping of untrained objects 1 and 4 in Fig. 10. Each joint angle is normalized to [-0.9, 0.9]. The dotted lines indicate the training data when the large box was wiped in the upper figure, and the large bowl was wiped in the lower figure. The solid lines show the motion generation results when untrained objects 1 and 4 were wiped. Although both of the solid lines are close to the training data, it can be seen that the angles were adjusted flexibly according to the situation.

We also show the tactile sensor data recorded during wiping untrained object 1 and 4 in Fig. 11. As shown in the upper figure, when the robot wiped untrained object 1, there were three peaks in the trajectory. In addition, the force values between the two peaks were 0 [N]. This result shows that switchback was done to wipe the cube. In contrast, although there were also some peaks in the force when the robot wiped untrained object 4, at least one sensor recorded 0.1 [N] or more at all times. This result shows that the robot wiped the object continuously.

We conducted the experiments using other untrained objects in the same way, and the results are summarized in Table IV. The success rate was calculated according to the wipe rule. Unfortunately, when the robot wiped untrained object 2, which has a triangular shape, the robot only touched the top of the object. Everything else was successfully wiped.

B. Experiment 2

We used trained objects to test changing the input data. As shown in Table V, the rate of the proposed input (i.e., Motor angle + Image feature + Tactile sensor data) was the highest.

When we input the motor angle and image features to the model, the success rate of wiping the upper and right sides in particular was low. In many of the failed cases, once the robot hand left the object, it could not measure the distance and instead floated above the surface. This shows that force feedback is important making contact with the surface and wiping constantly.

When the inputs were motor angle and tactile sensor data, the success rate of wiping the left and right sides of the big bowl and the left side of the small bowl were high, but others were extremely low. This is because the model was over-trained for only bowls. In fact, almost all the generated motions were similar to the bowl wiping method. Therefore, we can say that image information is necessary for recognizing the shape of the objects.

VII. CONCLUSION

In this research we proposed a DNN model for a robot to wipe 3D objects, considering "the wiping method." Previous research was limited to handling only 2D objects or needed computational models of the target in advance. We constructed a DNN model which consists of a CAE, which extract low-dimensional image features, and a MTRNN, which learns sensorimotor data dynamics. By using this model, the robot automatically recognizes the shape of an object and wipes it according to image and force information that are sensed in real time. We confirmed that the robot could generate wiping motions on untrained objects. In addition, we confirmed that both image and force information were necessary for recognizing and the objects to be wiped.

To our knowledge, this is the first work that uses only learning image, force, and arm joint information to enable a robot to trace unknown 3D shapes. It is the first step toward realizing a robot capable of wiping various objects.

In terms of wiping, thinking "where to wipe" is important. In future work, we will update our model to recognize the location of dirt on objects and plan efficient wiping trajectories. We will also update it to consider the amount of the force to apply according to the hardness of the dirt.

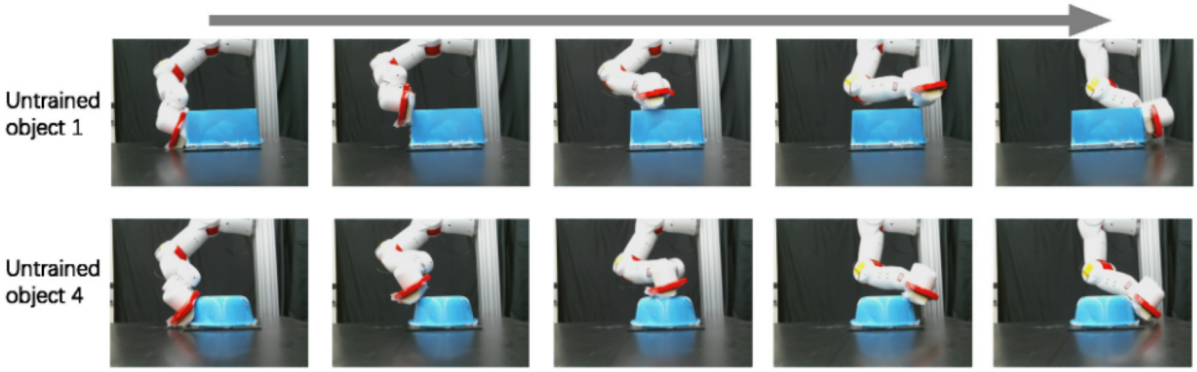

Fig. 9. The robot generates motions by the DNN model and wipes untrained objects.

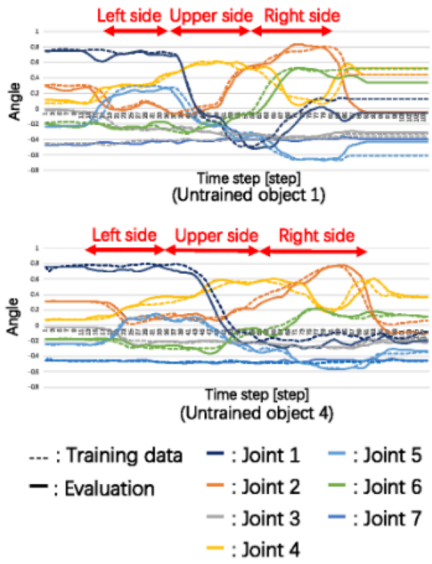

Fig. 10. Arm angle trajectory during wiping the untrained objects

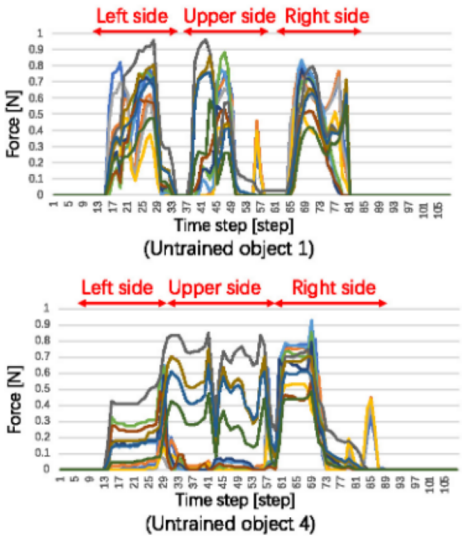

Fig. 11. Tactile sensor data during wiping the untrained objects

REFERENCES

[1] K. Yamazaki, R. Ueda, S. Nozawa, M. Inaba et. al. "Home-Assistant Robot for an Aging Society," Proceedings of the IEEE, Vol. 100, No. 8, pp. 2429–2441, 2012.

[2] M. Cakmak and L. Takayama, "Towards a comprehensive chore list for domestic robots," 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Tokyo, 2013, pp. 93–94.

[3] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, no. July, pp. 504-507, 2006.

[4] Y. Yamashita and J. Tani, "Emergence of Functional Hierarchy in a Multiple Timescales Recurrent Neural Network Model: A Humanoid Robot Experiment," PLoS Computational Biology, Vol. 4, No. 11, 2008.

[5] Tokyo Robotics Inc., Torobo Arm, https://robotics.tokyo/products/torobo_arm/

[6] D. Leidner, W. Bejjani, A. Albu-Schaśffer and M. Beetz, "Robotic Agents Representing, Reasoning, and Executing Wiping Tasks for Daily Household Chores," in Proceedings of International Conference on Autonomous Agents Multiagent Systems, pp 1006–1014, 2016.

[7] N. Cauli, P. Vicente, J. Kim, B. Damas, A. Bernardino, F. Cavallo and J. Santos-Victor, "Autonomous table-cleaning from kinesthetic demonstrations using Deep Learning," in Proceedings of IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob), pp.26–32, 2018.

[8] D. Martínez, G. Alenyà and C. Torras, "Planning robot manipulation to clean planar surfaces," Engineering Applications of Artificial Intelligence, Vol. 39, pp.23–32, 2015.

[9] F. Sato, T. Nishii, J. Takahashi, Y. Yoshida, M. Mitsuhashi and D. Nenchev, "Experimental Evaluation of a Trajectory/Force Tracking Controller for a Humanoid Robot Cleaning a Vertical Surface," in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.25–30, 2011.

[10] A. Gams, M. Do, A. Ude, T. Asfour and R. Dillmann, "On-line periodic movement and force-profile learning for adaptation to new surfaces," in Proceedings of IEEE-RAS International Conference on Humanoid Robots, pp.560–565, 2010.

[11] J. Hess, G. D. Tipaldi and W. Burgard, "Null space optimization for effective coverage of 3d surfaces using redundant manipulators", in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1923–1928, 2012.

[12] F. Nagata, T. Hase, Z. Haga, M. Omoto, K. Watanabe, "Cad/cam-based position/force controller for a mold polishing robot," Mechatronics, vol.17, no.4/5, pp.207–216, 2007.

[13] N. Saito, K. Kim, S. Murata, T. Ogata, and S. Sugano,"Tool-use Model Considering Tool Selection by a Robot using Deep Learning,"in Proceedings of IEEE-RAS International Conference on Humanoid Robots, pp.814-819, 2018

[14] P. J. Werhos, "Backpropagation through Time: What It Does and How To Do It," in Proceedings of the IEEE, vol.78, no.10, pp.1550–1560, 1990.

[15] Touchence Inc., *ShokacPot*^{TM} Product Outline, http://www.touchence.jp/en/cube/index.html