Tool-Body Assimilation Model using Neuro-Dynamical System for Acquiring Representation of Tool Function

Yuki Yamaguchi†, Harumitsu Nobuta†, Shun Nishide†, Hiroshi G. Okuno†, and Tetsuya Ogata‡

Abstract— **This paper proposes a model of tool-body assimilation with recurrent neural networks. Though there are existing models for this phenomenon, almost all the model requires the predefined model of the tool shape and/or active sensing process like grasping to recognize the tool. Our model enables a robot to acquire the representation of the motion dynamics of body with multiple tools, and to recognize the tools from the visual image of the tools without any knowledge of tools. Our model is composed of four modules. These modules are designed to perform respectively: image feature extraction, body dynamics learning, tool dynamic feature learning and tool recognition. A Self-Organizing Map (SOM) is used for extracting image features from raw images. Multiple Time-scales Recurrent Neural Network (MTRNN) is used as the body dynamics learning module. For tool feature learning, Parametric Bias (PB) nodes are attached to the neurons of MTRNN to modulate the behavior of MTRNN on the basis of the tool. Hierarchical neural network is used for recognizing tools from their shapes. The generalization capability of neural networks enables the model to deal with unknown tools. Experiments are conducted with ACTROID with multiple tools. Analysis of experimental results showed that tool dynamic features were acquired in PB space through training of the model. Motion generation experiments showed that the tool-body assimilation model can recognize tools from their shapes and handle them.**

I. INTRODUCTION

In recent years, the necessity of tool use ability has gained much attention for robots expected to be introduced into human society. The general goal of this paper is to create a basic model for robot's tool use on the basis of the way humans develop tool using abilities.

As a key idea to realize for robot to use tools, we adopt recent findings in cognitive fields, namely "tool-body assimilation". Tool-body assimilation represents the phenomenon humans feel as if their bodies extended to the edge of tool they use. This phenomenon emerges through tool use experiences and is said to be an important factor for tool use. Iriki et al. reported that tool-body assimilation can be observed with monkeys [1]. They recorded neurons named "bimodal neurons" before and after monkeys were trained to use a tool. Bimodal neurons respond both to tactile stimulation in the hand and to visual stimulation. Through tool use training, the visual receptive field of bimodal neurons expands from the monkey's hand to the surroundings of the grasping tool. This result shows that tool-body assimilation occurs on the nerve level. In this paper, we define tool-body assimilation as learning to modulate body model and generate motion based on the grasping tool through tool use experience. For tool use, we focus on reaching and object-moving with rigid tools. Beck proposed extension of reach is one of four factor for which animals use tools[2].

We deal with two problems in constructing the tool-body assimilation model for a robot system.

1) Design of tool features.
2) Recognition of tool function from the shape.

In the first problem, most existing studies on robots' tool use designs features for representing tools beforehand. Although these works were highly efficient in dealing with specific tools, they had difficulty dealing with unknown tools. Tool features should be acquired through tool use experiences so that they can adapt to various tools. In the second problem, we defined the tool function as the dynamic property of the tool. Inexperienced infants tend to rely on dynamic touch to determine the tool function [3]. Through tool manipulation experience, humans learn to presume a tool function from its shape [4]. Although there exist some models for tool-body assimilation, almost all the model needs dynamic touch to recognize tool function. It would be efficient if robots could learn to recognize tool function from its shape through tool use experiences like humans can. Thus, we challenge this problem.

We take the following approaches for these problems.

1) Self-organization of tool features from the dynamics observed during tool manipulation.
2) Mapping acquired tool dynamic features to image of tool (tool shape) with neural network.

In this paper, neuro-dynamical system, Multiple Time-scales Recurrent Neural Network (MTRNN), is used for representing a forward-inverse model of a robot body. In the first approach, we utilize a Self-Organizing Map (SOM) for extracting image features from camera images and Parametric Bias (PB) nodes, which represent tool dynamic features, are added to the body model (MTRNN). PB nodes are trained through the robot's active sensing experiences (i.e. tool use experiences). In this process, the necessary features are self-organized and need not be predefined. In the second approach, we utilize a hierarchical neural network for mapping PB nodes to tool shape. The robot can therefore presume tool function from the shape.

The rest of the paper is composed as follows. Section II discusses related works. Section III describes the overview of the model. Section IV explains the tool-body assimilation process using the model. Section V gives the experimental setup. Section VI details the experimental results. Section VII discusses the results. Section VIII presents conclusions

†Graduate School of Informatics, Kyoto University
‡School of Fundamental Science and Engineering, Waseda University

and future work.

II. RELATED WORKS

There are some studies on modeling tool-body assimilation.

Nabeshima et al. constructed the tool-body assimilation model based on robotics and the findings in cognitive fields[5]. In their work, the robot determined the tool inertia parameters through dynamic touch, and generated motion required to pull the object towards itself. Although the approach showed great effectiveness with various rigid tools, predefined tool features (inertia parameters) caused difficulties in adapting general tools and dynamic touch was required to determine the tool. In this paper, tool features are self-organized through tool using, and the robot learn to recognize tools from the appearance after training.

Stoytchev showed that the robot could acquire the affordance of tools through tool using[6]. In their work, the robot revised the affordance of tools when the target object did not move as expected and finally could adapt to unknown tools. Although his method was effective for a variety of tools, color information was used for determining the tools and many trials of tool manipulation were required for determining the affordances of unknown or broken tools. In our model, trials are not needed for recognizing known and unknown tools after training.

Hikita et al. used a saliency map to model tool-body assimilation phenomenon[7]. The robot arm could recognize the edge of a grasping tool through random motions. The focus of their work was recognition of the tool position and they did not deal with tool manipulation.

Nakagawa et al. constructed the tool-body assimilation model using a neuro-dynamical system[8]. Their experiment showed a robot dealing with an unknown tool. Although the method was effective, some trials were required to recognize the grasping tool. From the viewpoint of cognitive science, conversion of a self-body model based on tools in their method conflicts with cognitive knowledge. In this paper, we modulate rather than convert, as was done in Nakagawa's work, the behavior of self-body on the basis of tools.

III. OVERVIEW OF MODEL

In this section, we describe the overview of the tool-body assimilation model. The model is composed of four modules.

- image feature extraction module SOM
- body dynamics learning module MTRNN
- tool dynamic feature learning module PB-nodes
- tool recognition module Hierarchical Neural Network

Fig. 1 shows the overview of the model. A neuro-dynamical system, MTRNN [9], is trained for learning the relationships between visual features extracted using a Self-Organizing Map (SOM) [10] and motor sequences. By attaching PB nodes, which represent tool dynamic features, to self-body model (trained MTRNN), the behavior of MTRNN changes based on the values of PB nodes, which are calculated from a static image of a tool using a hierarchical neural network.

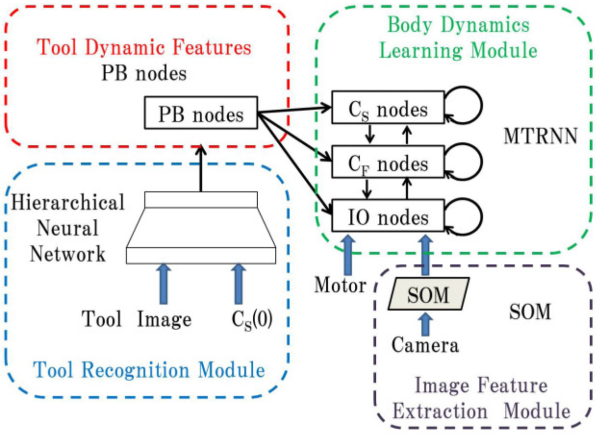


Fig. 1. Overview of Tool-Body Assimilation Model

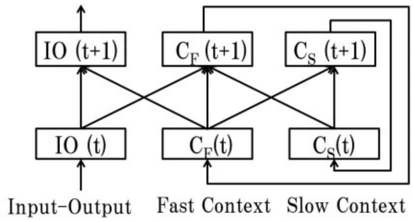


Fig. 2. Compositon of MTRNN

A. Image Feature Extraction Module (SOM)

Upon adaptation to unknown situations, image features should not be manually predefined. Therefore, we use a SOM for image feature extraction. We selected SOM because the compartibility of SOM and MTRNN was showed[11].

SOM is an unsupervised learning neural network developed by Kohonen [10]. SOM is composed of input and output layers. Neurons in the output layer are arranged two-dimensionally and possess weight vectors w with the same dimension as the input vector v.

In our model, the input vector is raw image pixels from the robot's camera. As the image features, we use the Euclidean distance of the weight and the input vector, $\parallel v - w \parallel$.

B. Body Dynamics Learning Module (MTRNN)

MTRNN is a version of the Jordan RNN [12], which can learn multiple sequential data and predict the next state $IO(t+1)$ from the current state $IO(t)$. The composition of MTRNN is shown in Fig. 2. MTRNN has two kinds of context nodes, fast context nodes (C_F) and slow context nodes (C_S). Nodes of MTRNN (IO, C_F, C_S) possess a time constant. Differences in time constants lead to differences in firing speeds of the nodes. The Back Propagation Through Time (BPTT) algorithm [13] is used for the training of MTRNN.

First, the outputs of the neurons are calculated recursively. The internal value of the ith neuron at step t, $u_i(t)$, is calculated as

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

τ_i : time constant of the ith neuron
$x_j(t)$: input value of the ith neuron from the jth neuron
w_{ij} : weight value from the jth neuron to the ith neuron
N : set of neurons connecting to the ith neuron

By applying the sigmoid function to internal value, the output of the ith neuron is calculated.

$$y_i(t) = \mathrm{sigmoid}(u_i(t)) = \frac{1}{1+\exp(-u_i(t))}. \quad (2)$$

The input value $x_i(t)$ is calculated using the output of the previous step $y_i(t)$, feeding back the teacher signal $T_i(t)$. The input values of the C_F and C_S nodes are equal to the output values of the previous step.

Second, weight is updated using the derivative of the training error. The training error E is calculated as,

$$E = \sum_t \sum_{i \in IO} (y_i(t-1) - T_i(t))^2 . \quad (3)$$

The weight from the jth neuron to the ith neuron is updated as,

$$w_{ij}(n+1) = w_{ij}(n) - \alpha \frac{\partial E}{\partial w_{ij}}. \quad (4)$$

α : training coefficient
n : update times

The initial C_S value, $C_S(0)$, is calculated using back propagation algorithm as,

$$C_{s_i}(0)(n+1) = C_{s_i}(0)(n) - \alpha \frac{\partial E}{\partial C_{s_i}(0)}. \quad (5)$$

After training, MTRNN can recover each sequence by the initial C_S value, $C_S(0)$. By inputting the $C_S(0)$, the output of step 0 is calculated. By recursively inputting the output back into the input, the whole sequence can be recovered.

C. Tool Dynamic Feature Learning Module (PB-nodes)

Connecting Parametric Bias (PB) nodes to each neuron of MTRNN, the robot generates motions with tools for training. The time constants of PB nodes are infinite, and values of PB nodes do not change during each sequence. Learning from differences in visual change due to differences in tool types, PB space learns to represent tool dynamic features. The weights from PB nodes to other neurons are updated the same way as the weights of MTRNN. The value of PB node is calculated by the same method as $C_S(0)$.

D. Tool Recognition Module (Hierarchical Neural Network)

We realize tool recognition from tool shapes, mapping the acquired tool dynamic features to the shape of the tool. Nishide et al. proposed a method for mapping dynamic features of objects to their static images by using Recurrent Neural Network with Parametric Bias (RNNPB) and a hierarchical neural network [14]. In their work, robots acquired the dynamic features of objects through active sensing and learned to estimate them from static images of objects. With tha same method as they suggested, we train a hierarchical neural network to map the PB space representing tool function to a static image of a tool and $C_S(0)$ representing each motion.

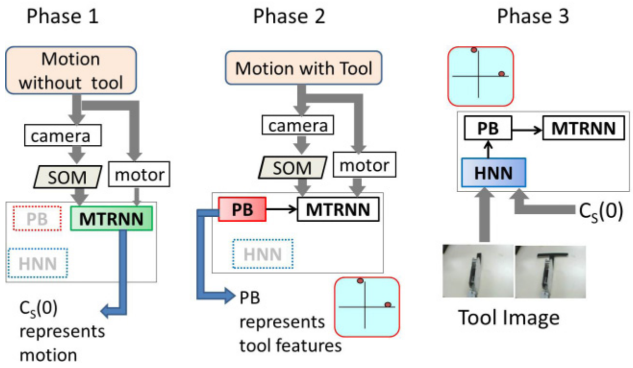


Fig. 3. Training Phases

IV. TOOL-BODY ASSIMILATION PROCESS

This section describes the tool-body assimilation process. Tool-body assimilation is conducted in the following five phases.

1) Train body model.
2) Self-organize tool dynamic features through tool manipulation.
3) Train tool recognition module.
4) Determine grasping tool from static image of the tool.
5) Generate motion with tool for given task.

The first three phases are done during training (shown in Fig. 3), and the other two are done when motion is generated (shown in Fig. 4).

A. Train Body Model

In the first phase, the robot executes predesigned motion patterns and interacts with a target object without a tool. During the robot motion, sequential image data are acquired from robot's camera. Features of these image data are extracted using SOM. Body dynamics learning module MTRNN is trained using the acquired image feature sequences and motor sequences. This phase links the robot motion (motor data) with the visual change in the image (image feature data), and $C_S(0)$ space acquired during training represents each motion pattern. This phase simulates human infant conducts random motions for recognizing own body.

B. Self-organize Tool Dynamic Features

During the second phase, the robot conducts the same motions as the first phase with some tools and sequential data (raw image and motor) are acquired as the same way as the first phase. By using the acquired data, the weights from PB nodes to MTRNN and values of PB nodes are trained with the weights of MTRNN fixed. In this phase, the tool dynamic features are self-organized in the PB space.

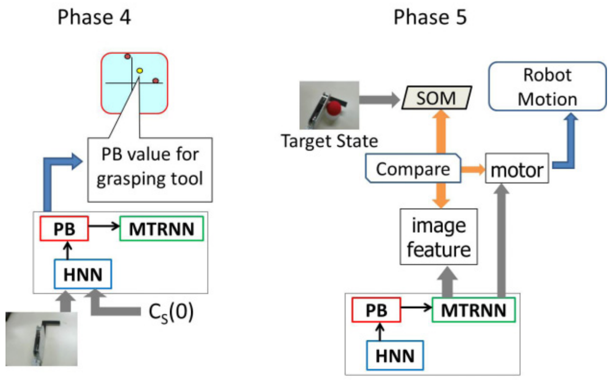


Fig. 4. Generation Phases

C. Train Tool Recognition Module

Hierarchical neural network is trained for mapping PB space acquired during the second phase to both a static image of the tool and $C_S(0)$ values representing each motion.

D. Determination of Grasping Tool

To determine the dynamical features of the grasping tool, the robot watches its hand and the tool. Hierarchical neural network outputs PB values from an observed static image of the tool and $C_S(0)$ acquired during the first phase.

E. Generate Motion Using Tool

Using PB values determined in the fourth phase and $C_S(0)$, the robot estimates changes in the visual image when conducting each motion. Given the target image, the robot compares the target image with the estimation and selects the most suitable motion for the given task.

V. EXPERIMENTAL SETUP

We used the humanoid robot ACTROID shown in Fig. 5 to evaluate our model. For tool use, we focused on object-moving operation. We used three tools (I-shaped, T-shaped, L-shaped shown in Fig. 6). The robot generated predesigned motions and manipulated a target object (a cylinder) bare hand or with tools (I-shaped and T-shaped) fixed to the left arm for acquiring training data. For evaluation of the model, we used the L-shaped tool as an unknown tool. The acquired data are composed of motor and image data.

1) Motor Data (7 dimensions)
During the experiment, the robot moved its left arm at seven angles to generate motions.

Fig. 5. Humanoid Robot ACTROID

Fig. 6. Tools used in Experiment

2) Image Data (768 dimensions)
Image data were acquired using the left camera of the robot with image resolution reduced to 32×24 pixels.

The input data (motor and image data) are normalized to [0.0, 1.0]. All data are acquired in 7.5 steps/sec for 30 steps.

Table. I shows the construction of the model.

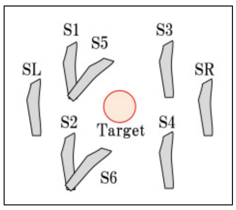
TABLE I
CONSTRICTION OF MODEL

Node Name	No. of Node	Time Constant
Motor Input Nodes	7	2
Image Feature Input Nodes	25	2
Fast Context Nodes	60	5
Slow Context Nodes	10	70
Parametric Bias Nodes	5	∞
Input Nodes for SOM	768	None
Input Nodes for Hierarchical NN.	778	None

A. Motion Patterns used in Experiment

Motion patterns the robot executed during training phase are predesigned. Each motion starts from a left or right initial state (SL or SR) and moves through two different postures. Both postures are determined by the position and direction of the robot's hand. Positional relationships between both postures and the target object are shown in Table. II.

TABLE II
HAND POSTURES USED IN MOTION PATTERN



State	Position	Direction
S1	far left	straight
S2	near left	straight
S3	far right	straight
S4	near right	straight
S5	far left	bend
S6	near left	bend

Thirty motion patterns were used for training. The robot generated each motions with T-shaped and I-shaped tools and bare hand. Therefore, we acquired 90 motion data for training.

- Left Init Motion (20 patterns)
  SL $\rightarrow$ S1 or S2 or S5 or S6 $\rightarrow$ S1$\sim$S6
- Right Init Motion (10 patterns)
  SR $\rightarrow$ S3 or S4 $\rightarrow$ S1$\sim$S6

B. Purpose of Experiment

In this paper, we assume that the robot should learn self-body dynamics before tool use training. Therefore, we conducted the following two training processes. By analyzing the PB space after training, we compared these two processes concerning the self-organization of tool dynamic features.

(a) proposed process: learn self-body (bare hand) with MTRNN and then train PB nodes through tool use with I-shaped and T-shaped tools.
(b) simultaneous learning process: learn all data (bare hand and each tools) at the same time with MTRNN attached PB nodes.

For evaluating the capability of tool recognition and manipulation, the robot was provided a static image of the tool and a goal state image. We examined what kind of motion the robot generated for each tool.

VI. EXPERIMENTAL RESULTS

A. Self-organization of Tool Function

Figure 7 shows the PCA compressed PB space after tool use training. The PB value is acquired for each sequence. The robot generated 30 types of motions for each tool, and therefore 30 PB values are acquired for each tool. The black circles represent PB values of robot motions without tools. The red squares and blue triangles represent PB values of motions with an I-shaped and a T-shaped tool, respectively.

Figure 7- (a) shows that in the proposed process, the distribution of PB values was clustered for each tool. From these results, the robot learned that the dynamic properties change with the grasping tool. On the other hand, Fig. 7-(b) shows that in the simultaneous learning process, the distribution of PB values was scattered and difference on the basis of tools did not appear.

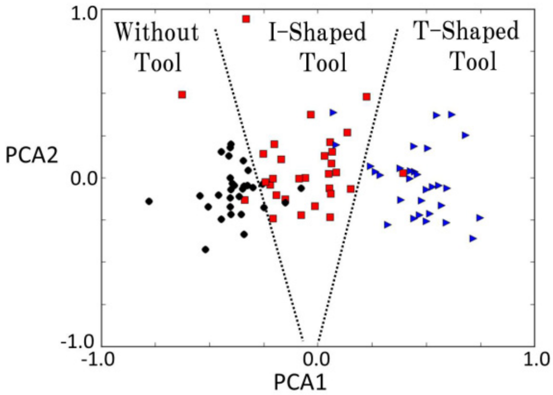
B. Motion Generation for Task

The robot determined the tool function by calculating PB values from a static image of the tool (shown in Fig. 8). The distribution of calculated PB values is shown in Fig. 9. Green rhombi, which represents PB values of an unknown L-shaped tool, seem to be distributed mainly between the PB values of the I-shaped tool and T-shaped tool.

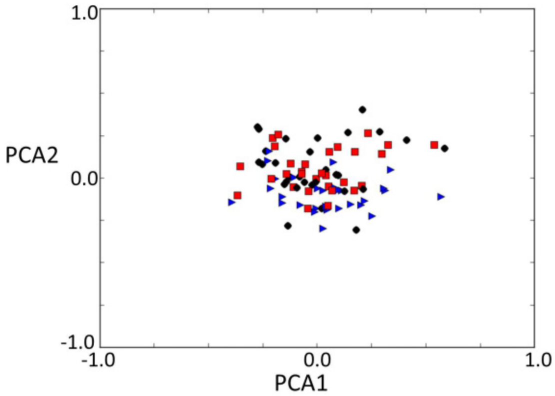
The goal state image was generated with the T-shaped tool as shown in Fig. 10. This image was acquired during training phase. The robot is required to retrieve the target object in front of the robot. The generated motions for the T-shaped tool, L-shaped tool, and bare hand are shown in Fig. 11, Fig. 12, and Fig. 13, respectively. The robot generated similar motions for the T-shaped and L-shaped tools. That motion is alike the SL → S1 → S6 motion described in Section V. For bare hand, the robot retrieved the target object by a pulling motion with its wrist bent (alike SL → S5 → S6 motion). These results show that the robot learned the function of the hooked shape to pull an object and reproduced the function with its hand.

VII. DISCUSSIONS

The difference in the distribution of PB values after tool use training denotes effectiveness of acquiring the forward-inverse model before tool use training. By learning in the proposed process, differences in observed dynamics based on tools are learned effectively and tool dynamic features self-organized. On the other hand, the distribution of PB values for simultaneous leaning process was not clustered for each tools. From this result, it cannot be said that the robot acquired the tool dynamic features and tool recognition module cannot be expected to generalize unknown tools. In this process, the network could not learn tool features effectively because it tried to learn changes based on motion patterns and changes based on tools together.



(a) proposed process



(b) Simultaneous learning process

Fig. 7. Self-Organized PCA PB Space of Tools

Fig. 8. Static Image of Tools for Tool Recognition

The results of the motion generation experiment shows the model could recognize tools only from their shapes and handle them. From the distribution of PB values calculated from static images of the tool, PB values of known tools are distributed similarly as in the training phase. PB values of an unknown L-shaped tool are distributed mainly between I-shaped and T-shaped tools. From this result, the model presumed the functionality of L-shaped tool changes between those tools on the basis of motion patterns. With the L-shaped and T-shaped tools, the robot retrieved the target using the hooked shape. In contrast, the robot retrieved the target by bending the wrist of its bare hand. This motion shows that the robot learned that the function of the L-shaped and T-shaped tools can be achieved by bending the wrist. These results imply the model's capability to perceive and generalize tool

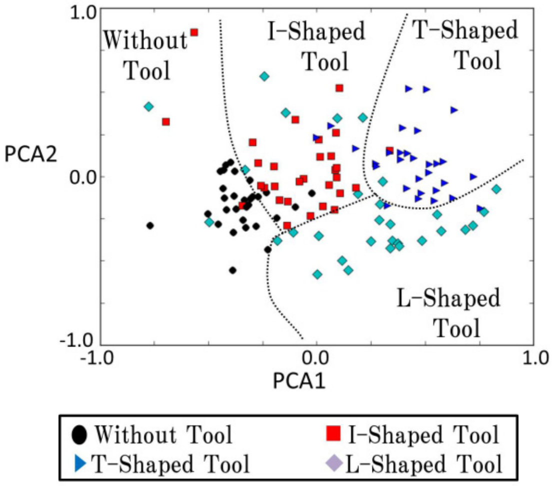


Fig. 9. PCA PB Space Recognized from Tool Shape

Fig. 10. Goal State Image

Fig. 11. Motion Generation with T-shaped Tool

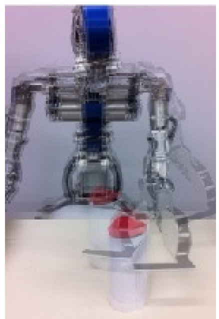


Fig. 12. Motion Generation with L-shaped Tool

Fig. 13. Motion Generation with Bare Hand

function.

Using our model, the robot could manipulate unknown but similar shaped tool (L-shaped tool) to those used at training phase (T-shaped tool). However it is difficult to deal with completely unknown tools. Humans learn to use unknown tools, based on estimation from tool shape and feed-backing the results. We believe estimation could make learning effective in that process. As the next step, we plan to develop our method so that the robot can learn completely unknown tools effectively in the same way as humans.

VIII. CONCLUSIONS

In this paper, we constructed tool-body assimilation model for a robot system. We conducted experiments using the humanoid robot ACTROID to evaluate the model. Analysis showed that tool features were acquired through training of the model. Motion generation experiments showed that the robot can recognize and handle tools including an unknown tool, by using tool shape as an only information.

For the next step, we plan to apply our model to a larger variety of tools and tasks. Furthermore, we plan to develop our method for manipulating completely unknown tool, which are not similar to tools used in the training phase, by feedbacking the results.

ACKNOWLEDGMENTS

This research was supported by JST PRESTO "Information environment and humans", the Ministry of Education, Science, Sports and Culture, Grants-in-Aid for Scientific Research (B) (21300076), (S) (19100003), and Creative Scientific Research (19GS0208).

REFERENCES

[1] A. Maravita and A. Iriki, "Tools for the body (schema)," *Trends in Cognitive Sciences*, vol. 8, no. 2, pp. 79–86, 2004.

[2] B. B. Beck, "Animal tool behavior: The use and manifacture of tools by animals," *New York: Garland STMP Press*, 1980.

[3] A. Streri and J. Feron, "The development of haptic abilities in very young infants: From perception to cognition," *Infant Behavior and Development*, vol. 28, pp. 290–304, 2005.

[4] C. F. Michaels, Z. Weier, and S. J. Harrison, "Using vision and dynamic touch to perceive the affordances of tools," *Perception*, vol. 36, pp. 750–772, 2007.

[5] C. Nabeshima, Y. Kuniyoshi, and M. Lungarella, "Towards a Model for Tool-Body Assimilation and Adaptive Tool-Use," *in Proc. ICDL*, pp. 288–293, 2007.

[6] A. Stoytchev, "Behavior-Grounded Representation of Tool Affordances," *in Proc. ICRA*, pp. 3060–3065, 2005.

[7] M. Hikita, S. Fuke, M. Ogino, and M. Asada, "Cross-modal body representaion based on visual attention by saliency," *in Proc. IROS*, pp. 2041–2046, 2008.

[8] S. Nishide, T. Nakagawa, T. Ogata, J. Tani, T. Takahashi, and H. G. Okuno, "Modeling Tool-Body Assimilation using Second-order Recurrent Neural Network," *in Proc. IROS*, pp. 5376–5381, 2009.

[9] Y. Yamashita and J. Tani, "Emergence of Functional Hierarchy in a Multiple Timescale Recurrent Neural Network Model: A Humanoid Robot Experiment," *PLoS Computational Biology*, vol. 4, no. 11, p. e1000220, 2008.

[10] T. Kohonen, "Self-Organization and Associative Memory," *2nd Ed, New York, Springer*, 1988.

[11] H. Arie, T. Endo, T. Arakaki, S. Sugano, and J. Tani, "Creating novel goal-directed actions at criticality: A neuro-robotic experiment," *New Mathematics and Natural Computation*, vol. 5, pp. 307–334, 2009.

[12] M. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," *Eighth Annual Conf. of the Cognitive Science Society (Erlbaum, Hillsdale, NJ)*, pp. 513–546, 1986.

[13] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representation by error propagation," *in D. E. Rumelhart and J. L. McLelland, editors Parallel Distributed Processing (Cambridge, MA: MIT Press)*, 1986.

[14] S. Nishide, T. Ogata, J. Tani, K. Komatani, and H. G. Okuno, "Predicting Object Dynamics from Visual Images through Active Sensing Experiences," *Advanced Robotics*, vol. 22, no. 5, pp. 527–546, 2008.