

Predicting Object Dynamics from Visual Images through Active Sensing Experiences

Shun Nishide, Tetsuya Ogata, Jun Tani, Kazunori Komatani, and Hiroshi G. Okuno

Abstract—Prediction of dynamic features is an important task for determining the manipulation strategies of an object. This paper presents a technique for predicting dynamics of objects relative to the robot's motion from visual images. During the learning phase, the authors use *Recurrent Neural Network with Parametric Bias* (RNNPB) to self-organize the dynamics of objects manipulated by the robot into the PB space. The acquired PB values, static images of objects, and robot motor values are input into a hierarchical neural network to link the static images to dynamic features (PB values). The neural network extracts prominent features that induce each object dynamics. For prediction of the motion sequence of an unknown object, the static image of the object and robot motor value are input into the neural network to calculate the PB values. By inputting the PB values into the closed loop RNNPB, the predicted movements of the object relative to the robot motion are calculated sequentially. Experiments were conducted with the humanoid robot Robovie-IIs pushing objects at different heights. Reduced grayscale images and shoulder pitch angles were input into the neural network to predict the dynamics of target objects. The results of the experiment proved that the technique is efficient for predicting the dynamics of the objects.

I. INTRODUCTION

Affordance is a feature of an object or environment that implies how to interact with the object or feature. For example, a button affords the motion of pushing it, while a chair affords the possibility of sitting down on it. The ultimate goal of our work is to functionalize affordance perception to the robot's ability. As the first step towards this goal, we have developed a method to predict the dynamics of an object from static images relative to the robot's motions. The proposed method would reciprocally link the robot motion with static and dynamic object features. Further on, the predicted object dynamics could be evaluated to generate the robot motions.

Learning the dynamics of objects requires object recognition with the sense of "active sensing"[1]. Noda et al. integrated multiple static features (size, weight, and color images) for object recognition while grasping the object with its hands[2]. The work used a three-layered SOM (Self-Organizing Map[3]) taking into account only the static features, which entails quite an arduous task for applying the results to motion planning where dynamics of objects need to be regarded. Ogata et al. worked on recognizing unknown objects based on the dynamics they bear[4]. Their method

classifies unknown objects based on the generalization capability of the Recurrent Neural Network (RNN) trained by the robot's action with a variety of objects. Works conducted by Takamuku et al. integrate static and dynamic properties of the object for classifying the object into predefined labels[5].

In this paper, the authors propose a technique to predict the dynamics of unknown objects from static images. Previous works considering recognition of unknown objects, require tactile contact with the objects for extracting dynamics. However, as round objects tend to roll when pushed, there is a tight connection between static and dynamic features. The proposed technique self-extracts and links static object features and robot motion to dynamic object features during the learning phase. This process leads to prediction of dynamics from static features without the necessity to have contact with the object.

The proposed method consists of two neural networks, each trained independently. The method first trains an RNN using visual sensory data generated while the robot hits a variety of objects at different heights. The dynamics of each object, relative to the robot motion, is self-organized according to their similarity. Next, the relationship of the static image, robot motion, and object dynamics are acquired by training a hierarchical neural network. Using the generalization capability of the proposed method, the dynamics of an unknown object is estimated according to the static image and robot motion.

The rest of the paper is composed as follows. Section II describes the proposed method with an introduction of the recurrent neural network model. Section III describes the robotic configuration and experimental setup. Section IV describes the actual prediction experiment using the humanoid robot Robovie-IIs. Section V describes the overall discussion considering the experimental results. Conclusions and future works are written in Section VI.

II. LEARNING ALGORITHM

This section describes the method to link static images to dynamic features through active sensing. The method consists of two learning phases. The first phase uses the experiences of active sensing to self-organize the similarity of the object dynamics. The FF-model (forwarding forward model), also known as RNN with Parametric Bias (RNNPB) model, proposed by Tani[6] is used. During this phase, the similarities of the dynamics are mapped into the PB space. The second phase uses a hierarchical neural network to link the static images with object dynamics. The network

S. Nishide, T. Ogata, K. Komatani, and H. G. Okuno is with the Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Kyoto, Japan {nishide, ogata, komatani, okuno}@kuis.kyoto-u.ac.jp

J. Tani is with the Brain Science Institute, RIKEN, Saitama, Japan tani@brain.riken.jp

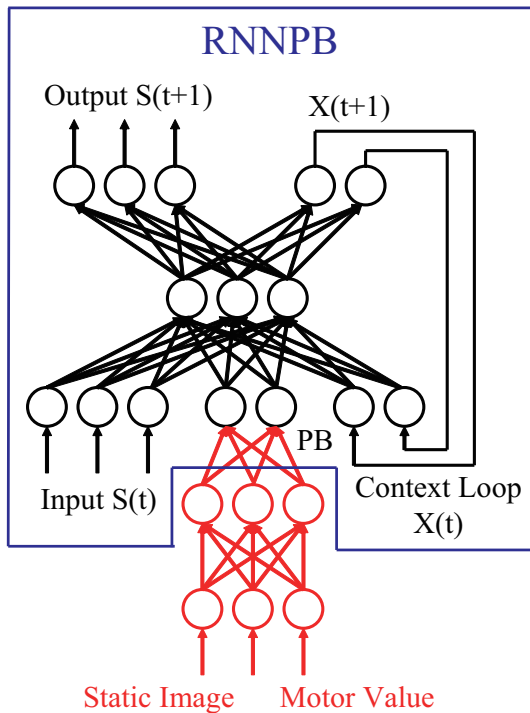


Fig. 1. Configuration of RNNPB and the Proposed System

configuration of the proposed method with RNNPB is shown in Fig. 1.

A. RNNPB Model

The RNNPB model is an extension to the conventional Jordan-type RNN model[7] with PB nodes in the input layer. It is capable of encoding multiple sequences into the RNN, altering its behavior according to the PB nodes.

The RNNPB is a supervised learning system requiring teaching signals as is the Jordan-type RNN model. The learning phase consists of weight optimization and self-organization of the PB values which encode each sequence using the back propagation through time (BPTT) algorithm[8]. The PB values are rearranged according to the similarity of the sequences creating clusters of similar sequences.

The RNNPB model possesses the capability to act as a predictor such that the next state could be calculated from the current state. Inputting the current state $S(t)$ and calculating the next state $S(t+1)$ as the output, a closed loop RNNPB is formed to calculate sequences recursively. The behavior of the predicted sequence varies with the PB values and the initial context state.

B. Learning Process

The PB values are learned in the same process as the BPTT algorithm. The back-propagated errors of the weights are accumulated along the sequences and used to update the PB values. Denoting the step length of a sequence as l , the update equations for the parametric bias at step t in

the sequence are as follows.

$$\delta\rho_t = k_{bp} \cdot \sum_{t-l/2}^{t+l/2} \delta_t^{bp} + k_{nb}(\rho_{t+1} - 2\rho_t + \rho_{t-1}) \quad (1)$$

$$\Delta\rho_t = \varepsilon \cdot \delta\rho_t \quad (2)$$

$$p_t = \text{sigmoid}(\rho_t/\zeta) \quad (3)$$

First, the δ force is calculated for updating the internal values of the PB p_t as (1). The calculation adds the accumulated delta error δ_t^{bp} with the low-pass filter which restrains rapid fluctuations of the PB values. The delta error is calculated by back propagating the errors from the output nodes to the PB nodes. Equations (2) and (3) represent the calculation method of the new PB by applying the sigmoid function to the internal value ρ_t updated using the delta force.

C. Two Phase Training

The proposed method contains an additional phase to the conventional RNNPB to link the static images to dynamics. Using the advantage that the RNNPB self-organizes the similarities of each sequence with numerical values, we attach a hierarchical neural network to the PB nodes.

The training phase of each neural network is conducted independently. First the PB values are self-organized using the method described in the previous subsection. Using the PB values as teaching signals, the hierarchical neural network is trained with static images and motor values as its input signals. The system, as a result, extracts the static features and movements of the robot that attracts most the dynamics of the object, and links them with the PB values of the RNNPB, which resemble the dynamics of the object.

III. ACTIVE SENSING EXPERIMENT

The authors have used the humanoid robot Robovie-IIs[9], which is a refined model of Robovie-II developed at ATR[10], for evaluation of the method. Robovie-IIs has 3 DOF (degrees of freedom) on the neck and 4 DOF on each arm. It also has two CCD cameras on the head for processing visual information.

A. Robot Motion and Target Objects

The shape of an object and robot motion are two large factors that affect the object dynamics. For instance, an upright box would tend to fall if pushed on the top, where it would just slide when pushed on the bottom. A cylindrical object would roll when pushed by its side.

The authors have focused on an active sensing motion that the robot pushes an object on the table with its arm. The pushing motion is conducted at different heights where the object and arm could take contact. Consequently, the objects are compelled to roll, fall over, or slide, depending on their shape and motion of the robot. Fig. 2 shows the scene of an actual experiment where Robovie-IIs pushes and moves the object by rotating the shoulder motor (roll axis). The procedure of the experiment is as follows.



Fig. 2. Robovie-IIs and Scene of Experiment

- 1) Acquire motion sequences from visual images while the robot pushes training objects.
- 2) Train RNNPB using the motion sequences.
- 3) Train the hierarchical neural network using the self-organized PB values, static images of training objects, and shoulder pitch angles.
- 4) Input static images of target objects and shoulder pitch angle into the hierarchical neural network to calculate PB values relative to the object and motion.
- 5) Input the PB values into the RNNPB for prediction.
- 6) Evaluate the predicted motion sequences.

7 types of objects were used for training the neural networks as shown in Fig. 3: 3 triangle poles, 3 cylinders, and 1 cuboid. A total of 17 motion sequences were acquired placing the objects in several orientations for manipulation. These motion sequences were used for training the RNNPB.

Prediction of motion sequences were conducted using 4 target objects shown in Fig. 4: a box, a shampoo container, a cylinder, and a packing tape. The cylinder, also used for training, was put upright for prediction, where it was laid down during training. The results of preliminary experiments with target objects considering the relationship between robot motion and consequence of the objects are shown in Table I. The alphabets correspond to the items of PB data illustrated



Fig. 3. Objects used for Training



Fig. 4. Targets used for Dynamics Prediction

TABLE I
ROBOT MOTION AND CONSEQUENCE OF OBJECT

Height of Robot Motion	High	Low
Box	Fall Over(a)	Fall Over(b)
Shampoo Container	Fall Over(c)	Slide(d)
Cylinder	Slide(e)	Slide(f)
Packing Tape	–	Roll(g)

afterwards in Fig. 5.

B. Configuration of the Neural Networks

The RNNPB used in this experiment is set as a predictor by calculating the next sensory state $S(t+1)$ as the output, from the current sensory state $S(t)$, the input. It consists of 28 neurons: 3 neurons each in the input/output layers, 10 neurons in the middle layer, 10 neurons in the context layer, and 2 neurons as parametric bias. As inputs of the RNNPB, the center position (x, y) and the inclination of the principal axis of inertia (θ) have been extracted from sequentially acquired images at 2.5 frames per second during the robot motion. These data are normalized $([0,1])$ before being input into the RNNPB.

The hierarchical neural network consists of $40 \times 30 + 1$ neurons in the input layer, 4 neurons in the middle layer, and 2 neurons in the output layer. The input is composed of a reduced grayscale image of the front view of the object (Resolution 40×30) and the target shoulder pitch angle (ϕ) which determines the height of the arm to push the object. These data are also normalized before being input into the neural network. The output layer is composed of the 2 PB values which encode the dynamics of the objects.

IV. EXPERIMENTAL RESULTS

A. Self-Organized PB Space and Calculated PB Values of Target Objects

The authors carried out the experiment using a total of 17 motion sequences as described in the previous section. The RNNPB was trained by iterating the calculation 1,000,000 times which required approximately 20 minutes using Xeon,

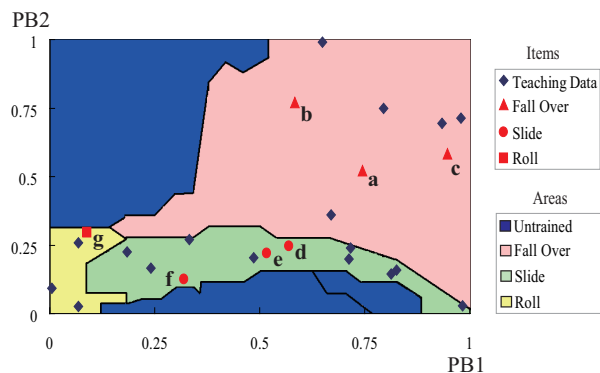


Fig. 5. Calculated PB Space

2.66 GHz. Each motion sequence consists of 7 steps, starting right before the robot arm has had contact with the object.

The RNNPB was trained using the 17 motion sequences of training objects. The PB values were self-organized according to the similarity of the dynamics. In other words, the PB values of the same object motions are allocated nearby. The PB space created by training RNNPB is shown in Fig. 5. The 17 teaching data are indicated as diamond-shaped marks. The remaining marks with numbers are the calculated PB values of target objects.

For analysis, the authors divided the PB space into 25×25 segments, and examined the object dynamics prediction of each area. Prediction is done by forming a closed loop RNNPB, feedbacking the output to the input. The motion sequence is calculated recursively from the initial position of the object. By evaluating the sequence, we labeled each segment into 4 motion categories (Untrained, Fall Over, Slide, and Roll). Untrained motions consist of a combination of 2 trained motions or a trained motion in a different direction. The four shaded areas represent clusters of each category.

After the PB values have been self-organized, the hierarchical neural network was trained using the (40×30) static images, shown in Fig. 6, and shoulder pitch angle. The calculation was iterated 30,000 times, which was decided heuristically to prevent over-training. The static images of target objects, shown in Fig. 7, and shoulder pitch angle for predicting the object motion were input into the neural network to calculate the PB values of the target object.

The calculated PB values of target objects are indicated as red marks in Fig. 5. The alphabets next to the marks correspond to the labels of target objects in Table I. The triangles, circles, and square each represent the distinction of dynamics, where they fell over, slid, and rolled. As can be seen, the PB values of the target objects reside in the area corresponding to their actual motions.

B. Motion Prediction from the PB Values

Using the closed loop RNNPB, the motion sequences of target objects are recursively predicted by inputting the PB values and initial context values, acquired during the training

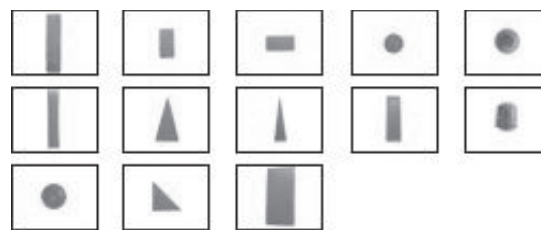


Fig. 6. Static Images of Training Objects

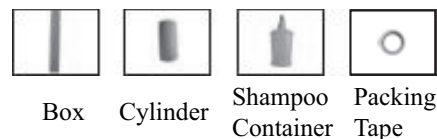


Fig. 7. Static Images of Target Objects

phase. The estimated sequences are shown in Fig. 8. Fig. 8(a), 8(b), 8(c) are motion sequences of objects that fell over. Fig. 8(d), 8(e), 8(f) are motion sequences of objects that slid. Fig. 8(g) is the motion sequence of the object that rolled. The seven predicted motion sequences represent well the actual motions indicated in Table I.

V. DISCUSSIONS

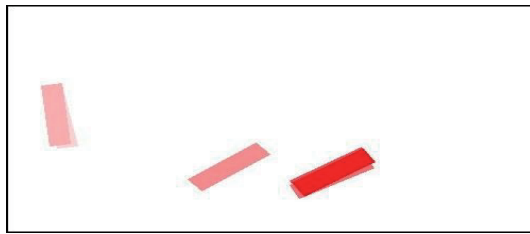
The proposed system combining two neural networks proved to be effective in predicting the dynamics of objects. As shown in Fig. 5, the PB values of target objects are distributed properly according to their actual motions. The predicted sequences in Fig. 8 express well the actual dynamics of the objects. In this section, we analyze the neural network to confirm the validity of the proposed system with some discussions about the results.

A. Analyzing the Hierarchical Neural Network

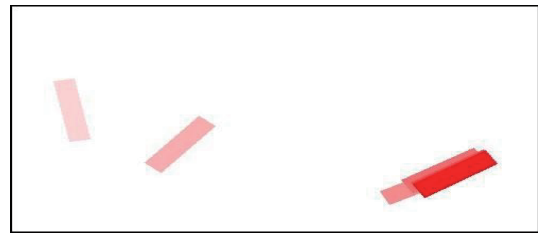
The hierarchical neural network functions as a filter to extract static features from images that affect the dynamics of objects. In this subsection, we investigate what types of features were extracted during the training process by analyzing the weights from the input layer to the middle layer of the hierarchical neural network.

The weights of the hierarchical neural network are presented in Fig. 9. We analyze the weights from the image input and robot motion input, each stated “Static Feature” and “Motion”, separately to evaluate the effects of each input. “Static Feature” is an image created by normalizing the weights to image format $([0, 255])$ for each of the image pixels.

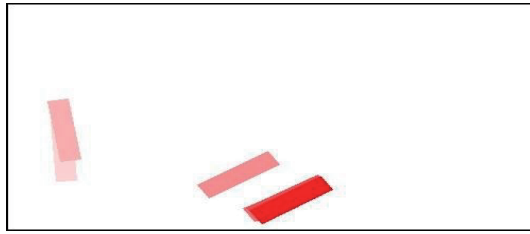
“Static Feature” represents filters for extracting the static features from object images. It is notable that these filters are created by combining portions of the input images of training objects in Fig. 6. Features that affect positively are displayed white, while those that affect negatively are displayed black. The images in “Static Feature” are applied to calculate the roundness and stability (fineness ratio, ratio of upper and lower surface of object) of the object, which



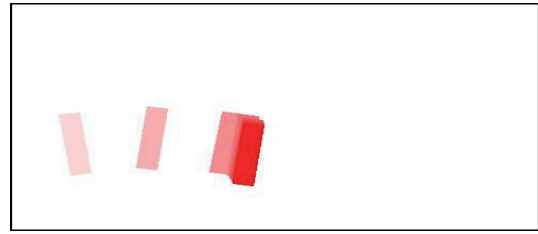
(a) Motion Sequence of Object a(Box, High)



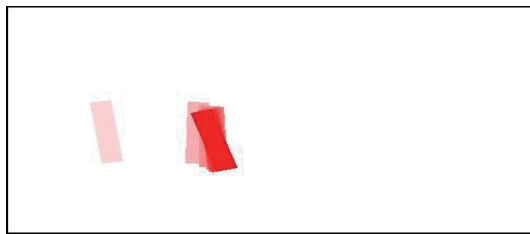
(b) Motion Sequence of Object b(Box, Low)



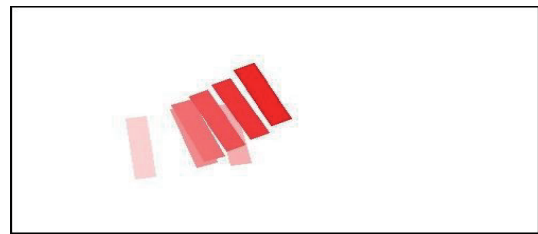
(c) Motion Sequence of Object c(Shampoo, High)



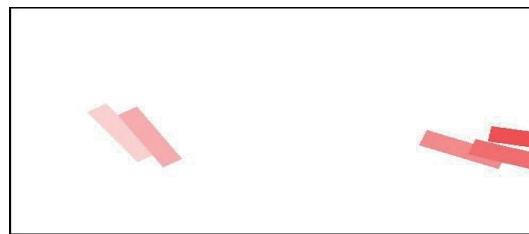
(d) Motion Sequence of Object d(Shampoo, Low)



(e) Motion Sequence of Object e(Upright Cylinder, High)



(f) Motion Sequence of Object f(Upright Cylinder, Low)



(g) Motion Sequence of Object g(Tape)

Fig. 8. Predicted Sequences of Target Objects

affects the dynamics (fall over, slide, roll) considered in the experiment. Features that induce the fall over motion of the objects are relatively colored black possessing a negative effect on the “Static Feature”. The edge of the objects are specifically intensified which denotes that the neural network has learned that the edge is a prominent static feature for predicting object dynamics.

“Motion” value appends a negative effect on “Static Feature”. The larger the magnitude, the larger the effect. This result implies that the neural network has learned that objects

would tend to fall over as the robot pushes the object at a higher spot.

We have analyzed the relation between the input and middle layers using the training and target images to investigate the qualitative roles of each middle node. Node 1 is a filter for calculating the stability of the object, which possesses high values for objects that slide and roll. Node 2, on the other hand, calculates the roundness of the object, having a high value for objects that roll. Node 3 functions as a filter to evaluate objects which have varying dynamics depending





Node	1	2	3	4
Static Feature				
Motion	-188.9	-153.7	-72.6	-143.6

Fig. 9. Weights of Hierarchical Neural Network

on the robot motion. This node generates high values for objects which would slide when pushed at the bottom, and fall over when pushed at the top. Taking into account that the magnitude of the “Motion” value is smaller for Node 3 than the other nodes, the decrease of this node value is smaller compared to the others when the robot motion increases. Therefore, the third node affects larger than the others when the robot pushes an object at a higher point. Node 4 adjusts the PB values and takes a high value for every object.

B. Dynamics Prediction using RNNPB

The experimental results proved that the proposed method is efficient in estimating the dynamics of objects with different shapes and sizes. The analysis has proved that the appropriate features were extracted for predicting the object dynamics. However, it is obvious that the neural network will predict cubes with approximately the same size as the cylinders as rolling objects, since the training process was done only with objects of different sizes. We would also like to investigate, what types of features would be extracted when the neural network is trained with similar sized objects.

A difficult issue in training the RNNPB is to decide the number of iterations for training. Large numbers of iteration would result in over-training which diminish the capability of generalization, while small numbers result in under-training. The training we have conducted created restricted space for the rolling motion. Although the neural network was capable of calculating an appropriate PB value for the packing tape, a wider area of the rolling motion in the PB space would result in better prediction for the motion sequence.

VI. CONCLUSIONS AND FUTURE WORKS

This paper proposed a dynamics prediction method through active sensing experiences combining the RNNPB and a hierarchical neural network. The method is composed of two phases, one for self-organizing the object dynamics by RNNPB and the other for linking static images and robot action to the self-organized dynamics using the hierarchical neural network. Training of the RNNPB was conducted with a total of 17 motion sequences, acquired from the pushing motion of the robot using seven objects placed in different orientations. Using the self-organized PB values, the hierarchical neural network was trained by inputting the reduced static images and shoulder pitch value. The hierarchical neural network, which links static images and motor values to PB values, proved efficient for extracting prominent static features that affect the dynamics of the object. By inputting the PB values to the closed loop RNNPB, the

motion sequences of unknown objects were predicted. The results showed that the dynamics of unknown objects could be predicted accurately. The analysis proved that the appropriate static features were extracted during the training process.

Our next goal is to apply the prediction results to manipulation strategies by increasing the number of robot motions. Thus, we could obtain higher generalization capability of the RNNPB and hierarchical neural network. This would lead to higher precision in the prediction process which is indispensable for manipulation strategies.

Prediction of a variety of motions resembles the possibility of actions the robot can take. By evaluating these possibilities, the robot could select the most appropriate action, in other words, the affordance. Reorganizing these actions relative to the static features, would lead to generalization of the affordance. We believe that these works would develop our system to a more sophisticated technique with large capabilities of application to studies in motion planning.

VII. ACKNOWLEDGMENTS

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (A)(No. 17680017, 2005-2007), and Kayamori Foundation of Informational Science Advancement.

REFERENCES

- [1] R. Bajcsy, “Active Perception,” *IEEE Proceedings, Special issue on Computer Vision*, Vol. 76, No. 8, pp. 996-1005, 1988.
- [2] K. Noda, M. Suzuki, N. Tsuchiya, Y. Suga, T. Ogata, and S. Sugano, “Robust Modeling of Dynamic Environment based on Robot Embodiment,” *IEEE ICRA 2003*, pp. 3565-3570, 2003.
- [3] T. Kohonen, “Self-Organizing Maps,” *Springer Series in Information Science*, Vol. 30, Springer, Berlin, Heidelberg, New York, 1995.
- [4] T. Ogata, H. Ohba, J. Tani, K. Komatani, and H. G. Okuno, “Extracting Multi-Modal Dynamics of Objects using RNNPB,” *IEEE/RSJ IROS 2005*, pp. 160-165, 2005.
- [5] S. Takamuku, Y. Takahashi, and M. Asada, “Lexicon Acquisition based on Behavior Learning,” *IEEE ICDL 2005, TALK 14*, 2005.
- [6] J. Tani and M. Ito, “Self-Organization of Behavioral Primitives as Multiple Attractor Dynamics: A Robot Experiment,” *IEEE Trans. on SMC Part A*, Vol. 33, No. 4, pp. 481-488, 2003.
- [7] M. Jordan, “Attractor dynamics and parallelism in a connectionist sequential machine,” *Eighth Annual Conference of the Cognitive Science Society*(Erlbaum, Hillsdale, NJ), pp. 513-546, 1986.
- [8] D. Rumelhart, G. Hinton, and R. Williams, “Learning internal representation by error propagation,” in *D.E. Rumelhart and J.L. McClelland, editors Parallel Distributed Processing* (Cambridge, MA: MIT Press), 1986.
- [9] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu, “Robovie: an interactive humanoid robot,” *Int. Journal of Industrial Robotics*, Vol. 28, No. 6, pp. 498-503, 2001.
- [10] T. Miyashita, T. Tajika, K. Shinozawa, H. Ishiguro, K. Kogure, and N. Hagita, “Human Position and Posture Detection based on Tactile Information of the Whole Body,” *IEEE/RSJ IROS 2004 Workshop*, 2004.