Analysis of Reliable Predictability based Motion Generation using RNNPB

Shun Nishide, Tetsuya Ogata, Kazunori Komatani, Hiroshi G. Okuno
Graduate School of Informatics
Kyoto University
Kyoto, Japan
Email: {nishide, ogata, komatani, okuno}@kuis.kyoto-u.ac.jp

Jun Tani
Brain Science Institute
RIKEN
Saitama, Japan Email: tani@brain.riken.jp

***Abstract*— Reliable predictability, which is tightly connected to consistency of environmental changes, is one of the main factors that determine human behaviors. As a constructive approach to understanding this mechanism, the authors have developed a method to generate autonomous object pushing motions based on consistency of object motions using a humanoid robot. The method consists of constructing a dynamics prediction model using Recurrent Neural Network with Parametric Bias (RNNPB), and motion searching based on an object consistency evaluation function using Steepest Descent Method. First, RNNPB is trained using the observed object dynamics and robot motion sequences, acquired during active sensing with objects. Next, the Steepest Descent Method is applied for searching the reliably predictable motion through the constructed dynamics model. Finally, the obtained motion is linked to the initial object image using a hierarchical neural network. The model inputs the object image outputting the reliably predictable robot motion which induces consistent object motions. The model was analyzed through two experiments, pushing cylindrical objects with a humanoid robot. The analysis has shown the method's effectivity and limitations to generate consistent object motions.**

I. INTRODUCTION

Motion generation based on affordance [1] is one of the key factors for providing environmental adaptability to robots. Affordance is a feature of an object or environment that implies how to interact with it. The two main streams of affordance based robot motion generations researches are conducted for mobility and manipulation.

Studies on affordance based mobility have progressed, granting robots the ability to determine traversable motions based on the current environmental state [2] [3]. These works apply active sensing [4] experiences with the environment to create a traversability knowledge of the environment. The knowledge is applied to classify the current environment (object) state into two groups, traversable or not traversable, selecting the traversable motion as the afforded motion.

Compared to affordance based mobility, only few works exist for affordance based manipulation. An example of such work is tool manipulation for extending the reach of the robot [5]. Extension of reach is one of the four factors that Beck proposes for which most animals use tools [6]. The work granted the robot the ability to autonomously select the appropriate tool for conducting an object manipulation task where the target object is out of reach.

Considering these backgrounds of affordance based motion generation researches, the authors focus on a more fundamental criteria. Referred to as *Reliable Predictability* by Hawkins [7], the capability to generate motions that produce predictable results is a prerequisite for robots to conduct a given task. The ability for robots to generate motions based on *Reliable Predictability* acts as a basis for the previous studies on affordance based mobility and manipulation to generate more robust motions.

Reliable Predictability is tightly connected to consistency of environmental changes. Humans are more capable of predicting consistent results than inconsistent ones. As *Reliable Predictability* is difficult to evaluate quantitatively, the authors evaluate consistency of environmental changes as a measure for *Reliable Predictability*. Hawkins also proposes that humans act to generate predictable, or consistent, results.

The aim of our work is to generate consistent object motion results from the object image based on the robot's active sensing experiences. The work contains two issues:

1) Creation of a model linking the dynamics of the object and robot.
2) Searching for robot motion that generates consistent object motions.

The authors have dealt with the first issue by applying the Recurrent Neural Network (RNN) model for learning object and robot dynamics. RNN possesses the capability to adapt to unknown environments with its generalization capability from few training data. For dealing with the second issue, the authors have applied the Steepest Descent Method for searching the most consistent object dynamics. The searching also yields the *Reliably Predictable* robot motion which generates consistent object dynamics, as the RNN links the robot and object dynamics. The final step links the initial object image to the acquired robot motion to create a model that inputs the object image, and outputs the *Reliably Predictable* robot motion. In a previous paper [8], the method had been proved to be capable of generating consistent rolling motions for cylindrical objects. In this paper, we analyze the method for two experimental setups, one presented in the previous paper and a different experimental setup.

A related work by Fitzpatrick aimed to generate rolling object motions, or to mimic an observed behavior [9]. In his work, the robot conducts active sensing with various objects to learn the ⟨object, action⟩ pair. The work applies motion generation to trained objects, whereas the authors utilize neural networks to apply to untrained objects. The objective of our approach also differs. While Fitzpatrick's approach generates goal-oriented behaviors, our approach generates the most predictable, or consistent, behavior.

The rest of the paper is composed as follows. Section II describes the proposed model. Section III briefly describes the setup and results of the experiments in the previous paper. Section IV describes the experiemental setup and results of the additional experiment. Section V presents some discussions considering the two experiments. Conclusions and future works are presented in Section VI.

II. Proposed Model

This section describes the overview of the proposed model. The model is composed of two neural networks, and three training phases. In the first phase, a robot-object dynamics model is created by training Recurrent Neural Network with Parametric Bias (RNNPB) [10]. A description on RNNPB is given in the following subsection. RNNPB is trained using robot motion and object sequences obtained during active sensing with training objects. In the second phase, the technique searches through RNNPB based on an object dynamics consistency evaluation function to acquire the *Reliably Predictable* robot motion. In the third phase, a neural network which links the initial object image with the acquired robot motion (PB) is trained. The construction of the model is shown in Fig. 1.

A. RNNPB Model

The authors utilize RNNPB, shown in the upper half of Fig. 1, as a model for learning and linking robot dynamics with object dynamics. RNNPB is an extension of the Jordan-type RNN [11], with Parametric Bias (PB) nodes in the input layer. It is set as a predictor calculating the next state $S(t+1)$ from the current state $S(t)$.

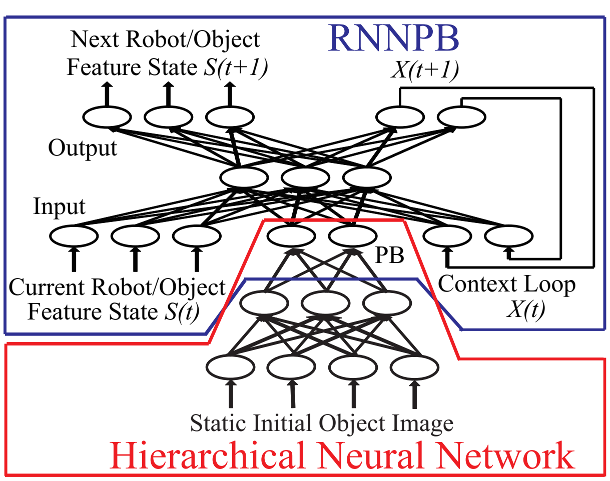

Fig. 1. Construction of Model

In comparison with the Jordan-type RNN, the PB nodes extend the ability of RNNPB to learn multiple sequential data in a single model. While RNN calculates a unique output from the input and context values, RNNPB is capable of altering its output by changing the values of the PB nodes (PB values). This provides RNNPB the capability to learn and generate multiple sequences. Therefore, RNNPB is often called a distributed representation of multiple RNNs.

The training process of RNNPB is conducted in a similar process as the Jordan-type RNN. Using teaching signals, the training optimizes the weights and self-organizes the PB values. The Back Propagation Through Time (BPTT) algorithm [12] is used for training. For updating PB values, the back-propagated errors of the weights are accumulated along the sequences. Denoting the step length of a sequence as T, the update equations for PB during the training phase are

$$\Delta\rho = \varepsilon \cdot \sum_{t=1}^{T} \delta_t^{bp}$$

$$p = sigmoid(\rho). \qquad (2)$$

First, the delta force $\Delta\rho$ is calculated by (1). The delta error δ_t^{bp} is obtained by back propagating the output errors from the output nodes to the PB nodes. The new PB value p is calculated by (2) applying the sigmoid function to the internal value ρ updated by the delta force $\Delta\rho$. ε is the learning constant.

During the training process of RNNPB, each training sequence is encoded into PB values based on their mutual similarities, forming the PB space which creates clusters of similar sequences. As the update equations for the PB values are conducted for each pattern sequences, a larger number of similar sequences creates a relatively wider cluster in the PB space. The sequences can also be reconstructed from the PB values by recursively inputting the output $S(t+1)$ back into the input $S(t)$. This process, called *association*, calculates the whole sequence from an initial state $S(0)$, initial context value $X(0)$, and a PB value.

B. Motion Searching based on Reliable Predictability

After training RNNPB with training sequences, the technique searches through the PB space for the most consistent object dynamics. The evaluation function is set as

$$E(p) = \frac{\delta O^2}{\delta p}, \qquad (3)$$

where O is the *associated* object dynamics and p is the PB value. Equation (3) evaluates the fluctuation of object dynamics relative to fluctuation of PB, which represents the robot motion. The local minimum of (3) indicates the PB encoding object dynamics with little deviation when the robot motion changes. In other words, the PB values of the local minimum encodes the most consistent object dynamics in its vicinity in the PB space. In this paper, the Steepest Descent Method is applied for calculating the local minimum.

For calculating (3), we discretize the function for numerical calculation, as it is difficult to be solved analythically. The function in (3) is equivalent to

$$E = \frac{1}{\mu}\sum_{i,j,t}(O(p_1, p_2, t) - O(p_1 + i\mu, p_2 + j\mu, t))^2$$

$$(i, j = -1, 0, 1) \quad (i \cdot j = 0), \tag{4}$$

where t is the step number of the sequence and μ is the discretization width. Equation (4) is expressed in two PB nodes, but a similar equation can be derived for larger number of PB nodes.

Obtaining the *Reliably Predictable* robot motion involves calculation of the most consistent object dynamics. As Steepest Descent Method is an initial value dependent method possessing many local minimums, we introduce another criteria for obtaining the robot motion. We evaluate the wideness of the PB space to determine a unique local minimum as the *Reliably Predictable* motion. As described in the previous subsection, a wider cluster is formed in the PB space for a larger number of similar patterns. As with humans, a larger number of experience provides better predictability of the environmental change. Therefore, the PB to be sought is the one with the largest number of points to converge from equally divided initial points. We divide the PB space defined in [0, 1] into lattice points, and use each lattice point as initial points to converge into a local minimum. The local minimum (PB) with the largest number of initial points to converge is the PB (p^*) encoding the robot motion which generates the most consistent object dynamics. The overview of the motion searching technique is shown in Fig. 2.

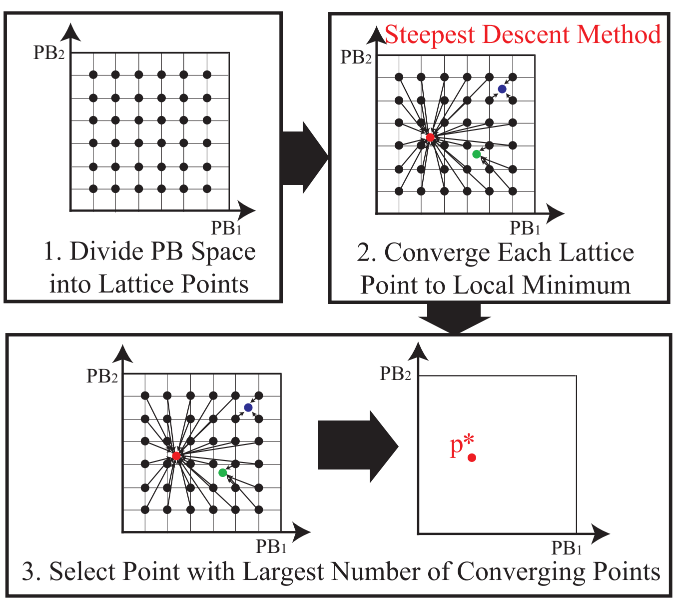

Fig. 2. Overview of Consistency Evaluation

C. Linking and Generating Motion from Object Image

The third phase consists of linking the *Reliably Predictable* robot motion (PB) acquired in the second phase, to the initial object image. We utilize a hierarchical neural network for linking the two. For motion generation, the object image is input into the hierarchical neural network to calculate the PB. The PB is then used to *associate* and generate the robot motion that induces a consistent object motion based on the robot's experience.

III. Motion Generation Experiment with Same Affordances

In this section, we describe the experiment conducted in the previous paper [8]. The authors have used the humanoid robot Robovie-IIs [13], shown in Fig. 3 for evaluation of the method. Robovie-IIs has three DOF (degrees of freedom) on the neck and four DOF on each arm. It also has two CCD cameras on the head for processing visual information, one of which was used in the experiment.

The actual experimental procedure was conducted as follows.

1) Acquire motion sequences of object features (center position and inclination of the principal axis of inertia of the object) from sequential images acquired while the robot pushes training objects.
2) Train RNNPB using motion sequences.
3) Search for consistent object dynamics using Steepest Descent Method.

The objects used for the experiment are shown in Fig. 4.

A. Training Data Acquisition and Model Training

In the previous paper, the model was evaluated with the pushing motion of Robovie II-s with cylindrical objects placed in the same initial position with different laid postures. As this experiment considers cylinders only in laid postures, every posture possesses the affordance to be rolled. The robot altered its shoulder pitch angle and elbow pitch angle to generate planar pushing motions with its left arm. The snack container and pen case, used for training, were laid in five postures. The robot pushed the objects with five motions acquiring a total of 50 motion sequences. During the training phase, a total

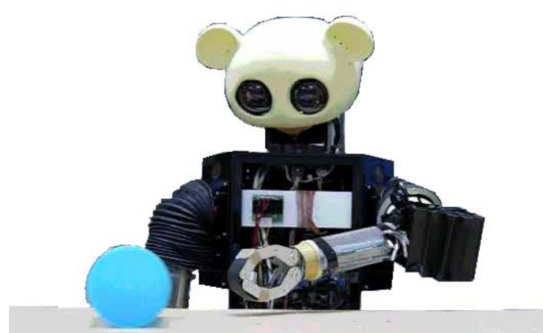

Fig. 3. Humanoid Robot Robovie-IIs

of 33 rolling motions were exhibited which were consistently generated when the robot pushed the cylinder along the shorter principal axis of inertia. No consistent motions were seen for other motions.

The configuration of the neural networks are shown in Table I and Table II. The input of RNNPB consists of the two robot joint angles, center point (2 DOF), and inclination of the principal axis of inertia (1 DOF) acquired at 2.5 frames/sec. For object dynamics consistency evaluation, the PB space was divided into 10×10 lattice points, converging each lattice point into a local minimum. The discretization width μ was set to 0.001. The input of the hierarchical neural network consists of a reducted grayscale image of the object with the resolution of 23×22. The inputs of the neural networks are normalized to [0, 1].

B. Results of the Experiment

Figure 5 shows the self-organized PB acquired during training and the PB values searched using Steepest Descent Method. The blue triangles and red squares each represent the PB values of rolling motions and other motions of training data, respectively. The black rhombuses represent the PB values calculated from the Steepest Descent Method in the second phase. As shown, it is notable that most of the rhombuses reside in the clusters of the PB values of rolling sequences, with a few deviations due to effects of the robot motion to the self-organization of PB. Using the trained model, object pushing experiments were conducted for the objects in Fig. 4, placing each object in five different postures. The pushing results of the experiments are shown in Fig. 6, generating a consistent rolling motion for every experiment.

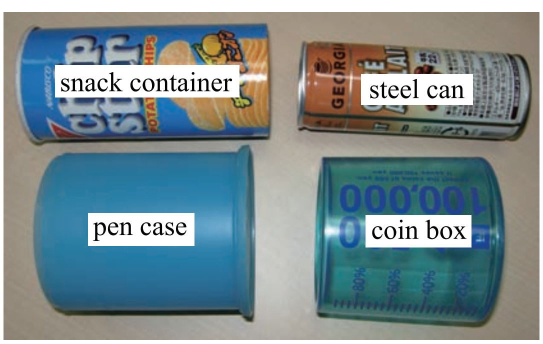

Fig. 4. Objects used for Experiment, Training Objects (Left) and Target Objects (Right)

TABLE I
CONFIGURATION OF RNNPB (PREVIOUS EXPERIMENT)

No. of Input/Output Nodes	5
No. of Middle Nodes	15
No. of Context Nodes	15
No. of PB Nodes	2
Learning Constant ε	0.03

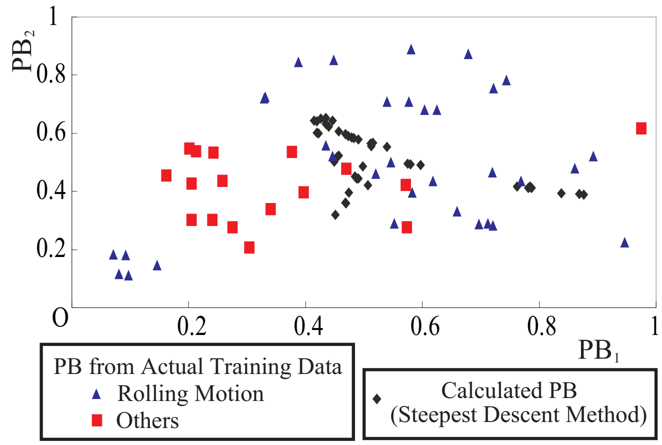

Fig. 5. Generated PB Space

IV. MOTION GENERATION EXPERIMENT WITH DIFFERENT AFFORDANCES

As an additional experiment, the authors have set up a different experimental environment from that described in the previous section. In the experiment described in the previous section, training data were acquired with the pushing motion of cylinders in laid postures. Therefore, every posture in the experiment contained only the affordance (motion possibility) of rolling the object. In the additional experiment, the authors evaluate the training process for objects with two different affordances. This experiment presents the limitations of the current method.

A. Experimental Setup

In this experiment, the snack container, pen case, and the steel can (Fig. 4) were used to acquire training data. The money box was neglected since it does not form a complete cylinder. As the setup for the experiment, the three objects were also put upright to generate falling over motions, in addition to the five postures considered in the previous experiment. The robot generated pushing motions by altering the shoulder roll angle, shoulder pitch angle, and the elbow yaw angle. Therefore, the robot pushed the objects from different angles at different heights.

From the data acquired during the robot's active sensing with objects, we use only those that generated the rolling motions and falling motions. When the object is put upright, the robot pushed the objects to generate falling motions. When the object is laid, the robot pushed the objects to generate rolling motions. Therefore, an upright object possesses the

TABLE II
CONFIGURATION OF HIERARCHICAL NEURAL NETWORK (PREVIOUS EXPERIMENT)

No. of Input Nodes	23×22
No. of Middle Nodes	10
No. of Output Nodes	2

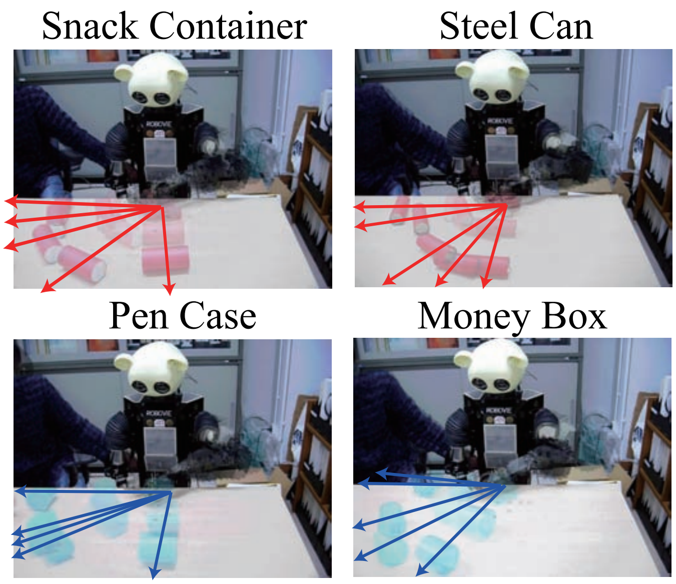

Fig. 6. Results of Object Pushing Experiments

affordance of falling over, while a laid object possesses the affordance of rolling. A characteristic of this experiment is that we consider two object states which have different motion possibilities, where the previous experiment considered one object state possessing two motion possibilities.

A total of 61 motion sequences were acquired, 52 for rolling and 9 for falling over. The total number of falling over motions are smaller compared to rolling motions since sequences with occlusions of the object are not used in the experiment. In this experiment, we used two cameras, each calculating the center position and principal axis of inertia of the object.

The configuration of RNNPB is shown in Table III. The input consists of three nodes for robot motor values and six nodes for object feature values. The rest of the configurations for the experiment are same as the previous experiment. In this experiment, we omitted the third phase training the hierarchical neural network, since the searched motions were scattered throughout the PB space.

B. Experimental Results

Figure 7 shows the PB acquired during training and the PB searched using Steepest Descent Method. For the PB of training sequences, the red rhombuses represent the PB of falling over sequences, while the blue squares represent the PB of rolling sequences. For the PB acquired using Steepest Descent Method, the green triangles represent those acquired

TABLE III

CONFIGURATION OF RNNPB

No. of Input/Output Nodes	9
No. of Middle Nodes	20
No. of Context Nodes	15
No. of PB Nodes	2
Learning Constant ε	0.01

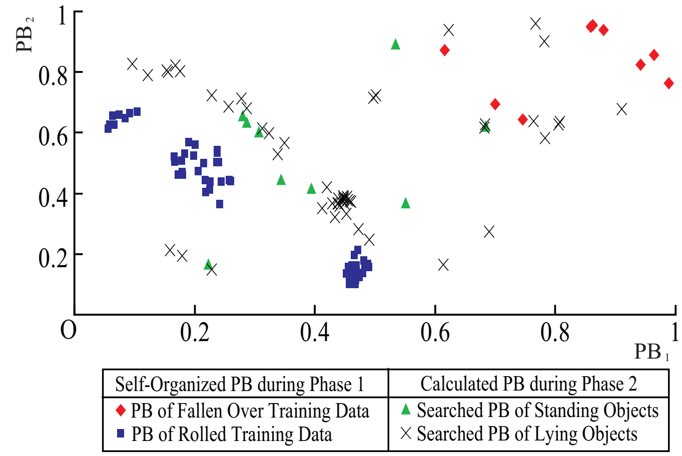

Fig. 7. Generated PB Space and Searched PB

from initially standing objects, while the black "X"s represent those acquired from initially laid objects.

From the results, it is notable that the method has not always succeeded in searching the reliably predictable motion. The green triangles and "X"s are inconsistently scattered throughout the PB space. This is due to the following two issues. First, training RNNPB creates two attractors (falling and rolling) with an discontinuous area between the two attractors. The neighborhood searching method proposed in this paper was unable to deal with the discontinuity of the PB space. Second, the model was unable to discriminate the two object states from the object features used in the experiment. This is due to insufficiency of object features to describe the object shapes. Therefore, the searching produced inconsistent results varying to small changes of the initial object feature. In order to adapt to different object states, the model requires extraction of appropriate object features that describe the object states.

V. DISCUSSIONS

In this section, we discuss the results of the experiments.

A. Predefined Object Features

In the experiments, the authors have used the center position and principal axis of inertia as object features for training RNNPB. As the principal axis of inertia is defined $[0, \pi]$, the value inverts when it reaches the limit. Figure 5 shows that the PB of rolling motions are divided into two clusters. This is due to the inversion of the principal axis of inertia. The motion searching algorithm also limits the variety of objects for the current model. Since the algorithm evaluates consistency from the initial object features (as the object dynamics are *associated* from the initial RNNPB input), it is necessary to select the appropriate object features that represent the object shape. In order to deal with these issues, an autonomous object feature extraction algorithm, which extracts object features that describe object dynamics, should be implemented to apply to different object shapes and dynamics.

B. Motion Searching with Different Affordances

In the first experiment, the robot was capable of generating consistent rolling motions of cylindrical objects. Although the self-organized PB values were not completely clustered (the rolling cluster was segregated into two parts due to insufficiency of object features), the PB values calculated by Steepest Descent Method in the second phase were obtained near the rolling motion clusters. However, in the second experiment, the method was incapable of searching for consistent object dynamics. This resulted from the following issues.

1) Limitation of the searching method to adapt to discontinuities in PB space.
2) Insufficiency of object features to discriminate object shape.

In order to resolve these issues, the model requires extraction of appropriate object features that describe the object dynamics. As described in the previous subsection, an autonomous object feature extraction algorithm is required to solve these issues. This would provide the model to adapt to a larger variety of objects.

VI. Conclusions and Future Work

In this paper, the authors described the analysis results of pushing motion generation with cylindrical objects using RNNPB. Training of the model consists of three phases. In the first phase, RNNPB is used to self-organize the robot-object dynamics data acquired during acitve sensing with objects. The second phase searches through RNNPB for the *Reliably Predictable* robot motion based on consistency of object dynamics. The third phase links the object image with the robot motion acquired in the second phase. The model therefore, inputs the object image and outputs the *Reliably Predictable* robot motion that generates consistent object motions.

Two experiments were conducted for analyzing the model. In the first experiment, the cylindrical objects were laid down during acquisition of training data altering its posture. Every object posture contained the affordance to generate the rolling motion in this experiment. The model was capable of generating consistent rolling motions adapting to the posture of the object. In the second experiment, the cylindrical objects were put in an upright posture, in addition to laid postures, to generate the falling over motion. In this experiment, the object contained the affordance to generate rolling motions or the falling over motions based on the posture of the object. This resulted in an inappropriate search of robot motion, as the model was incapable of distinguishing the difference of object states from the object features.

As future works, the authors plan to expand the model to a larger variety of objects. As the first step towards this goal, an autonomous dynamic object feature extraction based on active sensing experiences is necessary. This would provide the model the capability to deal with objects with different shapes. Then, we would consider extending the model to a hierarchical architecture. Thus, the model would be able to apply to a larger variety of objects and robot motions. Further on, the work could be integrated with previous studies on affordance based motion generation would improve the robustness of motion generation. The authors conclude this paper with a large expectation for affordance to be functionalized in the near future.

Acknowledgment

This research was partially supported by Global COE, the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (S), Grant-in-Aid for Young Scientists (A), Grant-in-Aid for Exploratory Research, RIKEN, and Kayamori Foundation of Informational Science Advancement.

References

[1] J. J. Gibson, *The Ecological Approach to Visual Perception*, Houghton Mifflin, ISBN: 0898599598, 1979.
[2] D. Kim, J. Sie, S. M. Oh, J. M. Rehg, and A. F. Bobick, *Traversability Classification using Unsupervised On-line Visual Learning for Outdoor Robot Navigation*, in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 518-525, 2006.
[3] E. Uğur, M. R. Doğar, M. Çakmak, and E. Şahin, *The learning and use of traversability affordance using range images on a mobile robot*, in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1721-1726, 2007.
[4] R. Bajcsy, *Active Perception*, in IEEE Proceedings, Special issue on Computer Vision, Vol. 76, No. 8, pp. 996-1005, 1988.
[5] A. Stoytchev, *Behavior-Grounded Representation of Tool Affordances*, in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3060-3065, 2005.
[6] B. B. Beck, *Animal Tool Behavior: The use and manufacture of tools by animals*, New York: Garland STMP Press, 1980.
[7] J. Hawkins and S. Blakeslee, *On Intelligence*, Times Books, ISBN: 0805078533, 2004.
[8] S. Nishide, T. Ogata, R. Yokoya, J. Tani, K. Komatani, H. G. Okuno, *Object Dynamics Prediction and Motion Generation based on Reliable Predictability*, to appear in Proceedings of the IEEE International Conference on Robotics and Automation, 2008.
[9] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, *Learning About Objects Through Action - Initial Steps Towards Artificial Cognition*, in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3140-3145, 2003.
[10] J. Tani and M. Ito, *Self-Organization of Behavioral Primitives as Multiple Attractor Dynamics: A Robot Experiment*, in IEEE Transactions on Systems, Man, and Cybernetics, Part A, Vol. 33, No. 4, pp. 481-488, 2003.
[11] 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.
[12] 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.
[13] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu, *Robovie: an interactive humanoid robot*, in International Journal of Industrial Robotics, Vol. 28, No. 6, pp. 498-503, 2001.