2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Object Dynamics Prediction and Motion Generation based on Reliable Predictability

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

Abstract— **Consistency of object dynamics, which is related to reliable predictability, is an important factor for generating object manipulation motions. This paper proposes a technique to generate autonomous motions based on consistency of object dynamics. The technique resolves two issues: construction of an object dynamics prediction model and evaluation of consistency. The authors utilize Recurrent Neural Network with Parametric Bias to self-organize the dynamics, and link static images to the self-organized dynamics using a hierarchical neural network to deal with the first issue. For evaluation of consistency, the authors have set an evaluation function based on object dynamics relative to robot motor dynamics. Experiments have shown that the method is capable of predicting 90% of unknown object dynamics. Motion generation experiments have proved that the technique is capable of generating autonomous pushing motions that generate consistent rolling motions.**

I. INTRODUCTION

Recently, motion generation based on affordance [1] is attracting many researchers' attentions. Affordance is a feature of an object or environment that implies how to interact with the object or environment. Currently, most works on affordance are performed for mobility, which generate motions based on traversability [2] [3]. These studies evaluate the traversability of an object to classify the robot's motions into two groups, traversable or not traversable, to select a traversable motion as the afforded motion.

As compared to affordance based mobility, only few works exist for affordance based object manipulation, despite that it is fundamental to human life. Stoytchev has addressed extension of reach as a criteria for generating tool manipulation motions [4]. This is one of the four factors that Beck proposes for which most animals use tools [5]. However, a more fundamental criteria exists for object manipulation: The object must be handled as predicted. This capability is referred to as *Reliable Predictability* by Hawkins [6], a neuroscientist who proposes that perception and behavior are based on predictability of environmental changes. The authors focus on *Reliable Predictability* as a criteria to generate motions.

Reliable predictability is tightly connected to consistency of environmental changes. Humans are more capable of predicting consistent results than inconsistent ones. It is predictable that a door with a doorknob will open when pushed/pulled on the doorknob. A plain door with no feature contains no predictable information since it may open either from the left or right. Therefore, 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 manipulation results from the object image based on the robot's active sensing [7] experiences. The work contains two issues:

1) Creation of a model to predict object dynamics from robot motions.
2) Evaluation of consistency for object dynamics.

The authors have dealt with the first issue by using a Recurrent Neural Network (RNN) for learning the object and robot dynamics. As robots possess hardware limitations (moving the robot too much would lead to damage of hardware), it is necessary for the robot to adapt to unknown environments from few training data. The generalization capability of RNN is one of the capabilities that meet this requirement. For dealing with the second issue, the authors have used the prediction model to search for the most consistent object dynamics relative to small variations in robot dynamics. The detected object-robot dynamics is linked to the initial object image through a hierarchical neural network. This provides the capability to generate consistent motions from the object image for known and unknown objects. In this paper, the authors have evaluated the prediction capability of the model and motion generation capability of the technique using the pushing motion of a humanoid robot.

A related work conducted by Fitzpatrick trains the robot to generate rolling motions of objects or to mimic an observed behavior [8]. The robot is trained using various objects learning the ⟨object, action⟩ pair for motion generation. The main difference of our approach is in the generalization capability of neural networks which enables the robot to generate motions for untrained objects. The objective of our approach also differs. Fitzpatrick's approach generates goal-oriented behaviors, where our approach generates the most predictable, or consistent, behavior.

The rest of the paper is composed as follows. Section II describes the proposed model and technique. Section III describes the experiment using a humanoid robot. Section IV describes the results and discussions for the prediction experiment. Section V describes the results and discussions for the motion generation experiment. Conclusions and future works are presented in Section VI.

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

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

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

II. Overview of Technique

This section describes the overview of the model and technique.

A. RNNPB Model

The authors utilize Recurrent Neural Network with Parametric Bias (RNNPB) [9] shown in the upper half of Fig. 1 for learning dynamics. RNNPB is an extension to the Jordan-type RNN [10] which contains Parametric Bias (PB) nodes in the input layer. In order to deal with sequential data (dynamics), RNNPB is set as a predictor which calculates the next state $S(t+1)$ from the current state $S(t)$.

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

RNNPB is a supervised learning system requiring teaching signals as is the Jordan-type RNN. The training phase consists of weight optimization and self-organization of PB values using back propagation through time (BPTT) algorithm [11]. 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

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

$$p = sigmoid(\rho). \tag{2}$$

First, the delta force for updating the internal values of PB p is calculated by (1). The delta error δ_t^{bp} in (1) is calculated 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 which is updated using the delta force. ε is a learning constant.

Training of RNNPB self-organizes the PB of each sequence according to their similarities, forming the PB space which creates clusters of similar sequences. In other words, the training encodes the sequences into PB values. Therefore, the sequences could 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 $S(0)$ and a PB value.

B. Predicting Object Dynamics from Robot Motion

Training the model for prediction of object dynamics consists of two phases. The first phase trains RNNPB using object motion data acquired from active sensing experiences with training objects. The input of RNNPB is the current object feature and the output is the object feature in the next state. The second phase trains a hierarchical neural network using the initial object image and robot motion as the input, and self-organized PB value as the output. The construction of the system is shown in Fig. 1. The inputs of the system are object image and robot motion. The output is the predicted object dynamics.

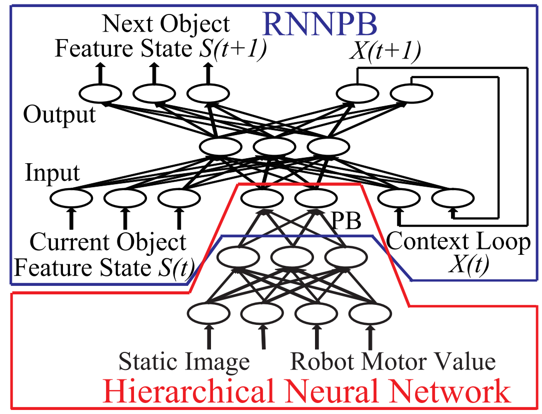

Fig. 1. Configuration of Prediction System

Prediction of environmental change is an output of a nonlinear function inputting various factors that affect the change. The model encodes object dynamics into PB values, and uses the hierarchical neural network as a nonlinear function which inputs object image and robot motion. For example, as a cylinder would roll when pushed by its side, the shape of an object (cylinder) and robot motion (push by its side) are two large factors that affect object dynamics (roll). Object shape is represented by raw object image, which is input into the hierarchical neural network. Object features used for training RNNPB are obtained from sequential object images acquired during active sensing. These features represent the description of the object dynamics itself. In this paper, object features input into RNNPB are predesigned as we focus more on the capability of the model to predict object dynamics. Ultimately, the input of the hierarchical neural network and RNNPB would be united as the object image.

Prediction of object dynamics is done by inputting only the initial frame of object image and robot motion into the hierarchical neural network to calculate the PB value. The PB value is then used to *associate* the whole sequence of object motion.

C. Motion Generation based on Object Dynamics Consistency

The model for generating motion based on object dynamics consistency is similar to the model in Fig. 1. The two differences are

1) Robot motion is trained using RNNPB, not hierarchical neural network.
2) There is an additional phase that searches for consistent object dynamics.

Considering the first difference, the motion generation system inputs the initial object image, and outputs robot motion. As introduced in the description of the prediction system, the input of the system is input into the hierarchical neural network and the output of the system is input/output into the RNNPB. Therefore the robot motion is shifted from the input of the hierarchical neural network to the input/output of RNNPB. This also provides the system to generate more dynamical motions compared to when the robot motion was input into the hierarchical neural network, since RNNPB is used for learning dynamical sequences. Therefore, the inputs/outputs of RNNPB are robot joint angle and object feature, and the input of the hierarchical neural network is the static image.

The first phase trains RNNPB using object/robot motion data acquired from active sensing experiences with training objects. The joint angle of the robot is input with the object feature into the RNNPB. The training forms the PB space, based on similarity of object/robot dynamics.

The second phase searches through the PB space based on a consistency evaluation function. The evaluation function is set as

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

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 (robot motion). The local minimum of (3), which indicates the PB encoding object dynamics with little deviation when the robot motion fluctuates, is sought by Steepest Descent Method. In other words, the PB of the local minimum encodes the most consistent object dynamics in its vicinity in the PB space.

For solving (3), we discretize the function for numerical calculation, as it is difficult to be solved analytically. The discretization of (3) derives

$$E = \frac{1}{\mu} \sum_{i,j,t} (O(p_1, p_2, t) - O(p_1 + i\mu, p_2 + j\mu, t))^2$$
$$(i, j = -1, 0, 1) \quad (i \cdot j = 0), \tag{4}$$

where t is the step number of the sequence and μ is the discretization width. Equation (4) is written for two PB nodes.

The PB to be sought is the one encoding the most consistent object dynamics. Steepest Descent Method is an initial value dependent method which possesses many local minimums. We evaluate the wideness of the PB space to determine a unique PB. During the training of RNNPB, consistent patterns are distributed widely in the PB space, as the update equations (1) and (2) are conducted for every pattern. Therefore, the PB to be sought is the one with the largest number of points to converge from equally divided initial points. In this method, 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 PB with the largest number of initial points to converge is the PB (p^*) encoding the robot motion which generates consistent object dynamics. The overview of the technique is shown in Fig. 2.

The third phase trains a hierarchical neural network to link the static object image to the PB value derived in the second phase.

For motion generation, the object image is input into the hierarchical neural network to calculate the PB value. The robot motion is *associated* by inputting the PB value into RNNPB.

III. Experimental Setup

The authors have used the humanoid robot Robovie-IIs [12] (Fig. 3), which is a refined version of Robovie-II [13], 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 authors have conducted two experiments using the pushing motion of the robot: one for evaluating the prediction capability and one for evaluating the motion generation capability. The training procedures of the experiments are as follows.

1) Acquire motion sequences of object features (center position and inclination of the principal axis of inertia of the object) from images while the robot pushes training objects.
2) Train RNNPB using motion sequences.
3) Search for consistent object dynamics. (for Motion Generation Experiment)
4) Train hierarchical neural network.

The objects used for the prediction experiment and motion generation experiment are each shown in Fig. 4 and Fig. 5, respectively.

A. Experiment 1 (Object Dynamics Prediction Experiment)

This experiment evaluates the prediction capability of the method with practical objects shown in Fig 4. The robot pushed the objects placed in various orientations at five different heights to generate rolling, falling over, sliding,

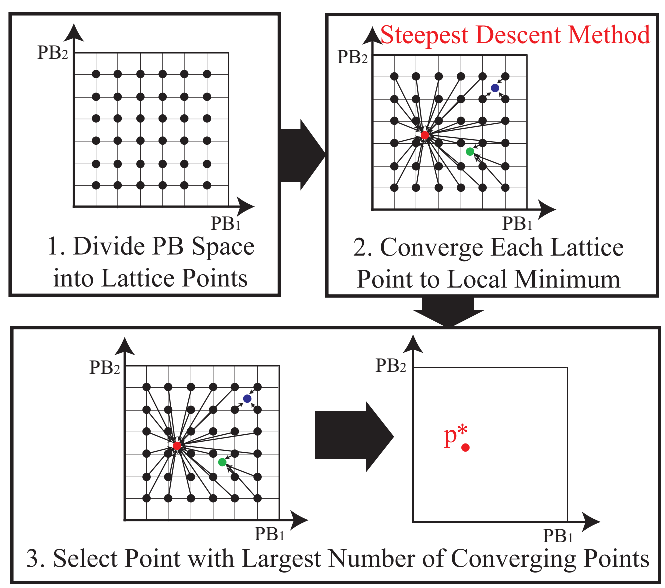

Fig. 2. Overview of Consistency Evaluation

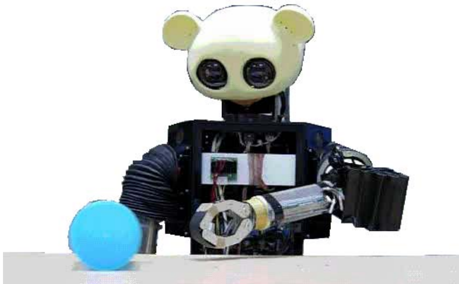

Fig. 3. Humanoid Robot Robovie-IIs

and bouncing motions of the objects. A total of 115 motion sequences were acquired (each consisting of 15 steps acquired at 10 frames/sec). 77 motion sequences were used for training the neural networks. The other 38 were used for prediction evaluation.

As the input/output for RNNPB, the center position (x, y) and the inclination of the principal axis of inertia (θ) of the object are used, extracted from sequentially acquired images. These data are normalized ([0, 1] for center position, [0.25, 0.75] for inclination of the principal axis of inertia) before being input into RNNPB. Considering the case of the ball, which has no principal axis of inertia, θ was set to 0.05. RNNPB was trained by iterating the calculation 1,000,000 times.

The input of the hierarchical neural network consists of the grayscale initial view of the object from the robot (Resolution 50×40) and the robot shoulder pitch angle (1 DOF). The background was eliminated before inputting the image. These input data are also normalized ([0,1]). The hierarchical neural network was trained by iterating the calculation 30,000 times.

The construction of the neural networks are shown in Table I and Table II.

B. Experiment 2 (Motion Generation Experiment)

This experiment consists of evaluation with cylindrical objects. The robot altered its shoulder pitch angle and elbow pitch angle to generate planar pushing motions. The snack container and pen case were each put in five orientations with the robot to push from five different angles to generate 50 motion sequences. During data acquisition, the objects generated consistent rolling motions when pushed along the shorter principal axis of inertia. A total of 33 rolling motions were exhibited, out of the 50 patterns. No consistent object motions were discovered for other robot motions. Therefore, the object dynamics to be generated for the experiment is the rolling motion of the object.

Fig. 4. Objects used for Prediction Experiment

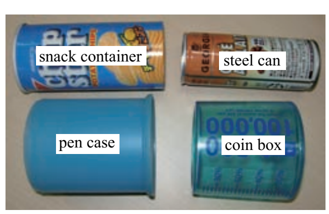

Fig. 5. Objects used for Motion Generation Experiment, Training Objects (Left) and Target Objects (Right)

The input of RNNPB consists of the same object features as the previous experiment, and the robot joint angles (2 DOF) acquired at 2.5 frames/sec. The robot joint angles are normalized ([0, 1]) before input into RNNPB. The RNNPB was trained by iterating the calculation 1,000,000 times.

For consistency evaluation, the authors divided the PB space into 10×10 areas. The inner 8×8 lattice points were used as initial points. The outer lattice points were neglected since the derivative may by miscalculated due to undefined area $(p_i < 0, p_i > 1)$. 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 (Resolution 23×22) acquired from the camera just before the robot pushes the object. The background was also eliminated as in the previous experiment. The images are normalized ([0,1]) before being input. The hierarchical neural network was trained by iterating the calculation 30,000 times.

The construction of the neural networks are shown in Table I and Table II.

For evaluation of motion generation the robot pushed the objects placed in the same initial position at random postures. The initial image of the object is input into the hierarchical neural network to calculate the PB, which is used to generate the motion.

IV. Results of Prediction Experiment (Experiment 1)

This experiment evaluates the prediction capability of the technique with a variety of practical objects. Training was conducted with 77 motion sequences: 35 sliding motions, 14 rolling motion, 19 falling motion, and 9 bouncing motions. The unknown target sequences consist of 10 sliding motions, 4 rolling motions, 8 falling motions, and 16 bouncing motions. As the authors have examined a few prediction results with sliding, rolling, and falling motions in the previous work

TABLE I
Construction of RNNPB

	Experiment 1	Experiment 2
No. of Input/Output Nodes	3	5
No. of Middle Nodes	40	15
No. of Context Nodes	40	15
No. of PB Nodes	3	2
Learning Constant ε	0.01	0.03

TABLE II
Construction of Hierarchical Neural Network

	Experiment 1	Experiment 2
No. of Input Nodes	$50 \times 40 + 1$	23×22
No. of Middle Nodes	20	10
No. of Output Nodes	3	2

[14], in which all the prediction have been successful, a larger number of bouncing motions have been included into unknown target sequences.

A. Dynamics Prediction Results

Fig. 6, Fig. 7, Fig. 8 and Fig. 9 show some examples of successfully predicted falling, sliding, rolling, and bouncing motion sequences. The "association" process calculates the motion trajectory of the center postion (x, y) and inclination of the principal axis of inertia θ. If $\theta > 0.1$, a rectangle is formed using the predicted (x, y) and θ. In other words, the rectangle represents the predicted trajectory of the principal axis of inertia of the object. If $\theta < 0.1$, we assume the prediction as bouncing motion of a ball, and draw a circle around (x, y).

The authors compared the predicted object motions with actual object motions. Table III shows the number of successful prediction for unknown target sequences. As the model creates predictions based on trained data, it is incapable of accurately predicting the actual sequences. Therefore, the authors have set the success of each prediction as follows.

1) Slide : Stop after center position shifts right
2) Roll : Center position shifts continuously right
3) Fall : Inclination of principal axis of inertia rotates about 90^o
4) Bounce : Center position oscillates up and down while shifting right

Considering sequences used for training, every predicted sequence accurately corresponded to the actual sequence. From Table III, the technique was capable of predicting more than 90% of the total unknown target sequences. Misprediction for the sliding pattern was predicted as fall over, which is discussed in the next subsection. The two unsuccessful predictions for the rolling patterns resembled a combination of several dynamics.

B. Discussions

The results in Table III have shown that the prediction capability of the method is stronger with dynamical object motions, such as fall over or bounce. Here, we discuss the factors that had led to failure in prediction for sliding and rolling motions.

The most prominent factor is in the predefined object features. In this experiment, the authors have used the center point and inclination of the principal axis of inertia for simplicity of object dynamics description. However, the sliding motion and rolling motion possess fairly similar dynamical properties when these object features are used: the difference exists only in the magnitude of the movement of the center position. In this paper, the authors predefined object features to evaluate the prediction capability of the model. Future works contain automatic object feature extraction based on object dynamics generated during active sensing and the task which the robot is to perform.

Another factor exists in unobservable properties of the object. For example, Fig. 10 shows the initial image of the object where the robot mistakened the sliding motion as falling motion. The prediction was for a robot motion that pushes the very bottom of the object. It is notable that even a human would mistaken the object dynamics as falling over.

V. Results of Motion Generation Experiments (Experiment 2)

This section describes the results of motion generation experiments with cylindical objects. The authors performed 20 motion generation experiments with objects shown in Fig. 5: five experiments for each of the four objects placed in random postures.

A. Generated PB Space

Figure 11 shows the PB space formed by training the RNNPB. PB values of training data, which were self-organized during training, are indicated as red triangles and green squares. Red triangles indicate PB values for rolling sequences of the objects. Green squares indicate those of sequences other than rolling. The distributions of PB values

TABLE III
PREDICTION RESULTS FOR TARGET SEQUENCES

Object Motion	Successful Prediction	Total
Slide	9	10
Roll	2	4
Fall Over	8	8
Bounce	16	16
Total	35	38

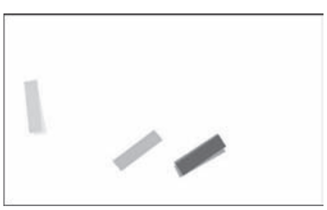

Fig. 6. Falling Prediction

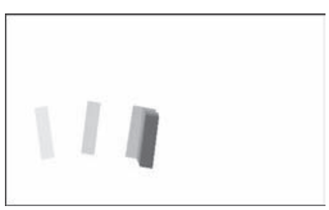

Fig. 7. Sliding Prediction

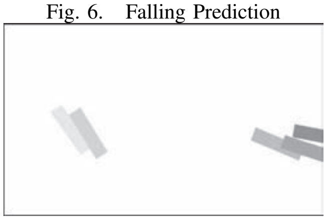

Fig. 8. Rolling Prediction

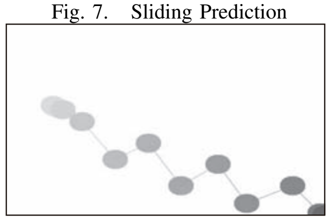

Fig. 9. Bouncing Prediction

Fig. 10. Initial Object Image for Sliding Prediction Failure

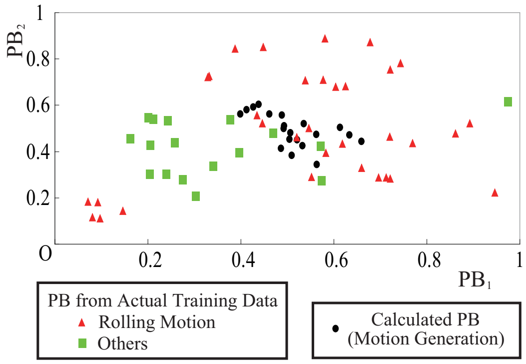

Fig. 11. Generated PB Space

are affected greatly by the number of times each motion has been observed as stated in Section II-C. Since the robot observed consistent (rolling) motions more than inconsistent ones, the distribution of PB values for rolling motions are wider compared to others.

Motion generation is done by first, calculating the PB value by inputting the initial image into the hierarchical neural network, and then *associating* the robot motion for generation. Black circles in Fig. 11 indicate the 20 PB values calculated by the hierarchical neural network during the experiment. It is notable that the circles reside mainly in the area where red triangles exist. These points are concentrated in the center of the PB space. This is a result of self-organization of PB values that the training has created a wide local minimum in the center of the PB space.

B. Generated Robot Motions

Figure 12 shows the results of the 20 experiments. (a)-(j) are motion generation results with training objects and (k)-(t) are those with target objects. The five arrows in each image

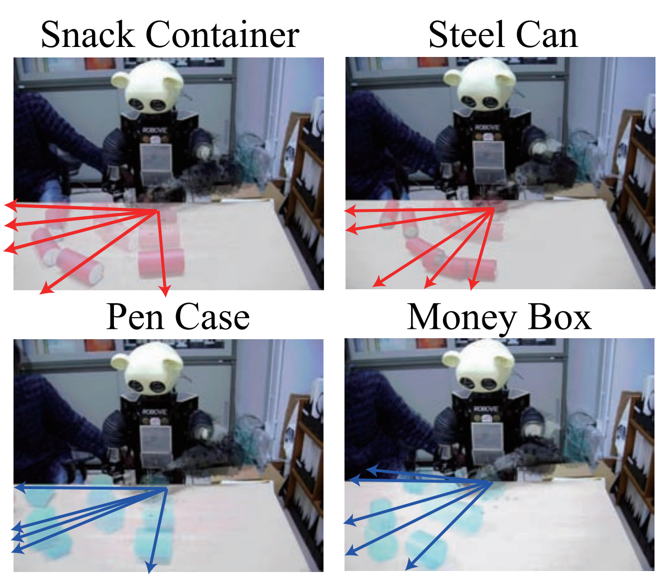

Fig. 12. Generated Robot Motions

represent the direction the object has rolled. From the results, it is notable that the method was capable of generating robot motions that would yield a consistent object rolling motion depending on the orientation of the object.

C. Discussions

This section presents the discussions considering the experimental results.

1) Predefined Object Features: Training RNNPB generates clusters of similar dynamics in the PB space. However, in this experiment, the PB values of rolling motions were bisected into two clusters. A major factor includes predefinition of object features as the center position and the inclination of the principal axis of inertia. The inclination of the principal axis of inertia is defined $[-\pi, \pi]$. Therefore it inverts when it reaches the limit. This has also resulted in an overlapping region of PB for rolling motions and other motions. Therefore, an automatic feature extraction method based on the robot's experience is required to achieve a more accurate description of object dynamics.

2) Perceived Affordance: In regard to real affordance [1] proposed by Gibson, which implies that the environment provides various action possibilities, perceived affordance [15] focuses on the actual behavior that one would take based on his/her experience. From the experiments, the robot generated pushing motions that yield consistent rolling motions of the object, adapting to the size and posture of the object. As consistency is closely related to *Reliable Predictability* the results denote that the model is capable of generating predictable motions. These results are due to active sensing experiences, where the robot has equally pushed the robot from various directions. Since the method evaluates the wideness of the PB space, the robot would generate a motion that the robot has observed more, when it randomly generates active sensing motions. While the authors have focused on *Reliable Predictability* as a criteria for generating motions, works exist which apply other criteria for motion generation as denoted in Section I. The results in this paper with other works imply the capability of the technique to functionalize perceived affordance to the robot's ability.

3) Scalability of the Method: We evaluate the scalability of the method based on three criteria.

(A) Scalability for Object Dynamics
(B) Scalability for Object Shape
(C) Scalability for Robot Motion

Considering scalability for object dynamics, the authors have shown that the model is capable of predicting Falling, Sliding, Rolling, and Bouncing motions. These motions cover most of the dynamics which objects can generate when being pushed. However, the current model uses the center position and inclination of principle axis of inertia as features to describe object dynamics. In order to deal with more complicated object motions, the model should be refined so that these features are automatically extracted to best describe object dynamics.

Considering scalability for object shape, the authors have used object image as input for the hierarchical neural network to give the robot capability to recognize the object shape. From the results of Experiment 1, the model was capable of predicting dynamics of various object shapes. Future works contain evaluation of the model's ability to generate motions with complicated object shapes.

Considering scalability for robot motion, RNNPB is capable of learning complicated robot dynamics as it could learn object dynamics. However, learning different motions (such as pushing and grasping) with the same model will associate the two motions in the PB space. This will also affect the motion searching phase, which should also be evaluated to express the effectivity of the method. In order to distinguish different motions, the model should contain a selective module in the upper level which selects the robot motion, and train different RNNPB for each motion. Creation and evaluation of such model is still left as future work.

VI. Conclusions and Future Works

This paper proposed an autonomous robot motion generation technique based on *Reliable Predictability*. As *Reliable Predictability* is tightly connected to consistency of environmental changes, the authors have evaluated consistency of object dynamics for generating motions.

The method utilizes RNNPB, Steepest Descent Method, and a hierarchical neural network to solve the two issues of the technique. The first issue, creating a prediction model, was resolved by first training the RNNPB with object dynamics, and then attaching a hierarchical neural network to link static object images and robot motion to object dynamics. The method was capable of predicting 90% of the total unknown sequences, denoting the effectivity of the method.

The second issue, evaluation of object dynamics consistency, was resolved by using the prediction model to associate the object dynamics, relative to robot motions. First, RNNPB was trained using robot and object dynamics. Next, a consistency evaluation function was set using the derivative of the object dynamics relative to PB. Steepest Descent Method was used to calculate the local minimum from lattice points. The method evaluates the spatial spread of the PB space, and selects the PB value with the largest number of lattice points to converge. Lastly, the PB value is linked with the initial object image. For motion generation, the object image is input into the hierarchical neural network to calculate the PB value. The PB value is input into RNNPB to *associate* robot motion. The robot was capable of generating robot motion that would yield consistent rolling motions for cylindrical objects, denoting the effectivity of the technique.

Future works include evaluation of the technique with other objects and robot motions, and an automatic feature extraction method for the input/output of RNNPB. In this paper, the center position and inclination of the principal axis of inertia was incapable of completely distinguishing object rolling motions from other motions, though the generalization capability made it possible to generate robot motion which yields object rolling motion. We plan to use the output errors of RNNPB to automatically extract dynamic object features based on the robot's experience. This would provide the capability to apply to more complex object shape and dynamics. Further on, we would move on to investigate the effectivity of the method with other robot motions. We believe that these works combined with related studies from the field would lead to functionalization of perceived affordance to the robot's ability.

VII. Acknowledgments

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, A. F. Bobick, "Traversability Classification using Unsupervised On-line Visual Learning for Outdoor Robot Navigation," in *Proc. ICRA*, pp. 518-525, 2006.
[3] E. Uğur, M. R. Doğar, M. Çakmak, E. Şahin, "The learning and use of traversability affordance using range images on a mobile robot," in *Proc. ICRA*, pp. 1721-1726, 2007.
[4] A. Stoytchev, "Behavior-Grounded Representation of Tool Affordances," in *Proc. ICRA*, pp. 3060-3065, 2005.
[5] B. B. Beck, "Animal Tool Behavior: The use and manufacture of tools by animals," New York: Garland STMP Press, 1980.
[6] J. Hawkins and S. Blakeslee, "On Intelligence," *Times Books*, ISBN: 0805078533, 2004.
[7] R. Bajcsy, "Active Perception," in *IEEE Proc., Special issue on Computer Vision*, Vol. 76, No. 8, pp. 996-1005, 1988.
[8] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, G. Sandini, "Learning About Objects Through Action - Initial Steps Towards Artificial Cognition," in *Proc. ICRA*, pp. 3140-3145, 2003.
[9] 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.
[10] 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.
[11] 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.
[12] 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.
[13] 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," in *IEEE/RSJ IROS Workshop*, 2004.
[14] S. Nishide, T. Ogata, J. Tani, K. Komatani, H. G. Okuno, "Predicting Object Dynamics from Visual Images through Active Sensing Experiences," in *Proc. ICRA*, pp. 2501-2506, 2007.
[15] D. A. Norman, "The Psychology of Everyday Things," *New York: Basic Books*, ISBN: 0465067093, 1988.