Paper:

Autonomous Motion Generation Based on Reliable Predictability

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

*Graduate School of Informatics, Kyoto University
Engineering Building #10, Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan
E-mail: {nishide, ogata, komatani, okuno}@kuis.kyoto-u.ac.jp
**Brain Science Institute, RIKEN
2-1 Hirosawa, Wako City, Saitama 351-0198, Japan
E-mail: tani@brain.riken.jp
[Received December 1, 2008; accepted May 29, 2009]

Predictability is an important factor for generating object manipulation motions. In this paper, the authors present a technique to generate autonomous object pushing motions based on object dynamics consistency, which is tightly connected to reliable predictability. The technique first creates an internal model of the robot and object dynamics using Recurrent Neural Network with Parametric Bias, based on transitions of extracted object features and generated robot motions acquired during active sensing experiences with objects. Next, the technique searches through the model for the most consistent object dynamics and corresponding robot motion through a consistency evaluation function using Steepest Descent Method. Finally, the initial static image of the object is linked to the acquired robot motion using a hierarchical neural network. The authors have conducted a motion generation experiment using pushing motions with cylindrical objects for evaluation of the method. The experiment has shown that the method has generalized its ability to adapt to object postures for generating consistent rolling motions.

Keywords: neurorobotics, neural networks, humanoid robots

1. Introduction

Intelligent connection of perception to action is the very definition proposed by Brady for intelligent robots [1]. Such robots require perceptual mechanisms to cognize the space, while linking these mechanisms to action. Conventionally, many motion generation studies have predefined the perceptual features of the environment and their connection to action. Therefore, such works were incapable of adapting to unknown environments for generating adaptive behaviors. As an alternative to this approach, many works currently focus on active sensing [2] for perceiving and generating motions for environmental adaptability.

An important concept for introducing intelligent robots into the human society is KUKANCHI. KUKANCHI (Japanese word for Intelligent Human-Space Design and Intelligence) is a term for creating an intelligent space that involves both humans and robots residing in it. From the perspective of KUKANCHI, two main approaches exist for linking perception and action. The first is to embed behavioral information into the environment for robots to acquire. This is a powerful approach that can be applied to well-defined environments. The robot would recover the embedded behavioral information to generate intelligent actions. The second approach is from the perspective from affordance theory [3], where motion information are embedded as affordance in the natural environment. Our approach adopts the second approach to extract behavioral information from the natural environment based on active sensing.

Works on active sensing were originally conducted for object classification and identification. Ogata et al. applied active sensing to extract dynamical multi-modal information [4]. They used a recurrent neural network model to classify objects based on the extracted information. The generalization capability proved the method's effectivity to deal with unknown objects. Takamuku et al. applied active sensing to relate the extracted dynamical information with static visual properties to label objects [5]. Their method utilizes reinforcement learning for learning and identifying object-oriented behaviors. Hebbian network is then applied to associate behavior with labels. These works only apply active sensing for object recognition, and motion generation is out of the scope.

Application of active sensing for motion generation have mostly been conducted for generating goal-oriented motions. Fitzpatrick et al. applied active sensing for learning the relation between robot motion and resulting object motion [6]. Stoytchev focused on tool affordance to relate the relation between robot, object, and tool [7]. Although these works have shown highly effective results, they contain two issues.

1. Lack of adaptability to unknown objects due to predefinition of object features.

2. Lack of generating adaptive robot motions to unknown objects due to usage of initially designed motions.

Fig. 1. Objects used for prediction experiment from the work of [8].

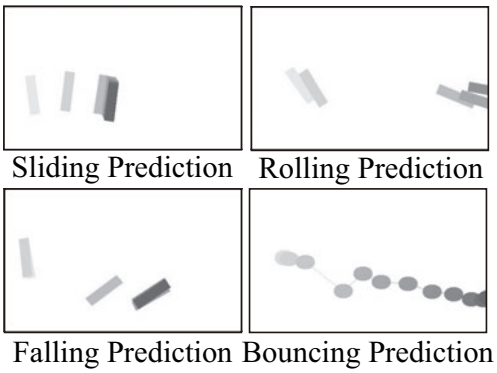


Fig. 2. Object motion prediction results from the work of [8].

Our research focuses on resolving these two issues, and creating a robot system that generates robot motions that could handle objects as predicted. We define such motions as motions with high *reliable predictability*.

In our previous work, we have dealt with the first issue [8]. We constructed an object dynamics prediction model using a recurrent neural network, as a dynamics learner, and hierarchical neural network, as a feature extractor. The input of the system is object image and robot motion, and the output is the predicted object dynamics. During the training process, the appropriate static object features that affect the object dynamics are self-organized within the hierarchical neural network. Therefore, the model was capable of adapting to various unknown objects (**Fig. 1**) to predict the object motions (**Fig. 2**) by applying the generalization capability of the neural networks.

In this paper, we present our approach to deal with the second issue, generating robot motions based on *reliable predictability*. Utilizing the model created in [8], we search through the model based on a reliable predictability evaluation function using steepest descent method. The searched motion is related to the object image through the hierarchical neural network. The generalization capability of the model is applied to generalize the motion generation process relative to object image.

From the cognitive science perspective, *reliable predictability* is an important factor in human behaviors [9]. Recognition is achieved when a change in an environment is predictable, and humans generate motions to acquire predictable results. Specifically in object manipulation, it is a prerequisite for the robot system to handle objects as predicted. As it is difficult to quantitatively evaluate *reliable predictability*, we evaluate the consistency of object motions, which is closely connected to *reliable predictability*. Under this assumption, we define reliable predictability as: *Object motion with small alteration relative to alteration of robot motion*. In other words, object motions are consistent when the same object motions can be observed for similar robot motions.

Our works lies as a basis to the works by Fitzpatrick and Stoychev [6, 7]. Goal-oriented tasks consist of a number of robot motion solutions that would fulfill the task. Some motion solutions are more reliable than others, based on the robot's body capability and experience. By combining our work with these previous works, the robot would be capable of selecting the most reliable motion out of the multiple solutions.

The rest of the paper is composed as follows. Section 2 describes the proposed model and technique. Section 3 describes the experiment using a humanoid robot. Section 4 describes the results and discussions for the prediction experiment. Section 5 describes the results and discussions for the motion generation experiment. Conclusions and future works are presented in Section 6.

2. Overview of Technique

This section describes the overview of the technique for searching and generating reliably predictable motions. The method consists of three training phases with a preliminary phase. In the preliminary phase, the robot applies its active sensing motion to acquire robot motor sequences, object motion sequences, and initial object image. The initial robot motor value is fixed as described later in this section. In the first phase, the robot motor sequences and object motion sequences are used to train the dynamics learner. We utilize Recurrent Neural Network with Parametric Bias (RNNPB), proposed by Tani [10], as the dynamics learner. In the second phase, the steepest descent method is used to search through RNNPB for a reliably predictable robot motion (PB) based on an evaluation function. In the third phase, the robot motion (PB) searched in the second phase is linked to the initial object image using a hierarchical neural network. Phase 1 and Phase 2 are described in the following subsections. Therefore, the model links the object image with the reliably predictable robot motion through the three training phases. The configuration of the model is shown in **Fig. 3**. An overview of the whole method is shown in **Fig. 4**.

2.1. Phase 1: Training of RNNPB

We utilize Recurrent Neural Network with Parametric Bias (RNNPB) shown in the upper half of **Fig. 3** for learning dynamics. RNNPB is an extension to the Jordan-type

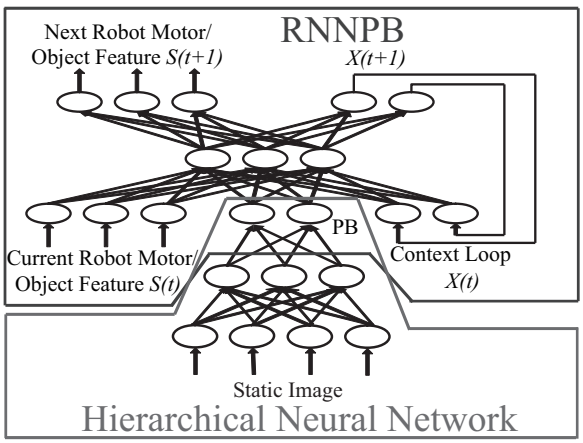


Fig. 3. Configuration of model.

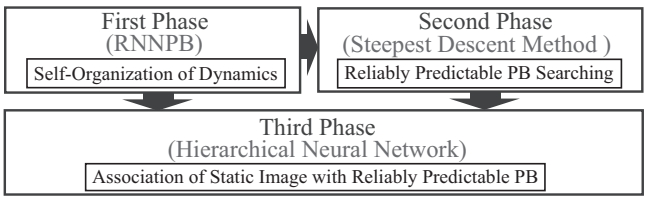


Fig. 4. Overview of model training.

RNN [11] 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 input/output consists of robot motor values and object feature values.

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 [12]. 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

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

$$p = sigmoid(\rho). \quad \ldots\ldots\ldots\ldots \quad (2)$$

First, the delta force for updating the internal values of PB p is calculated by Eq. (1). The delta error δ_t^{bp} in Eq. (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 Eq. (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 *closed loop calculation* calculates the whole sequence from an initial state $S(0)$, initial context value $X(0)$, and a PB value. *Open loop calculation* is a process that calculates the output by inputting the observed state value for $S(t)$ instead of the previous output $S(t-1)$.

2.2. Phase 2: Motion Searching Based on Reliable Predictability

Using RNNPB trained in the first phase, we search for the reliably predictable robot motion using steepest descent method in the second phase, based on an evaluation function. Under the definition of *Reliable Predictability* as *Robot motion that generates consistent object motions relative to small alteration in robot motion*, we set the evaluation function as

$$E(p) = \frac{\delta O^2}{\delta p} \quad \ldots\ldots\ldots\ldots \quad (3)$$

where O is the object dynamics calculated through *Closed Loop Calculation* and p is the PB value. Eq. (3) evaluates the fluctuation of object dynamics relative to fluctuation of PB. As altering the PB results in alteration of robot motion, this evaluation function evaluates the magnitude of the change of object dynamics from small alteration of robot motion. By minimizing Eq. (3), a local minimum PB can be acquired, which indicates the PB encoding object dynamics with little deviation when the robot motion fluctuates in the vicinity of the local minimum.

As data are acquired discretely, we discretize the function for numerical calculation. The discretization of Eq. (3) derives,

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

where t is the step number in the sequence, $O(p_1,p_2,t)$ is the object sequence calculated from the PB value (p_1,p_2) and t, and μ is the discretization width. Eq. (4) is written for the case of two PB nodes, but a similar equation can be derived for a larger number of nodes.

The capability of motion searching largely depends on the object features used for describing object motion. Calculation of $O(p_1,p_2,t)$ requires an initial robot motor value, initial object feature value, and initial context value, in addition to the PB value (p_1,p_2) and t. We fix the initial robot motor value for each pattern (which is not an impractical assumption as there is no problem in moving the robot to a fixed posture before starting manipula-

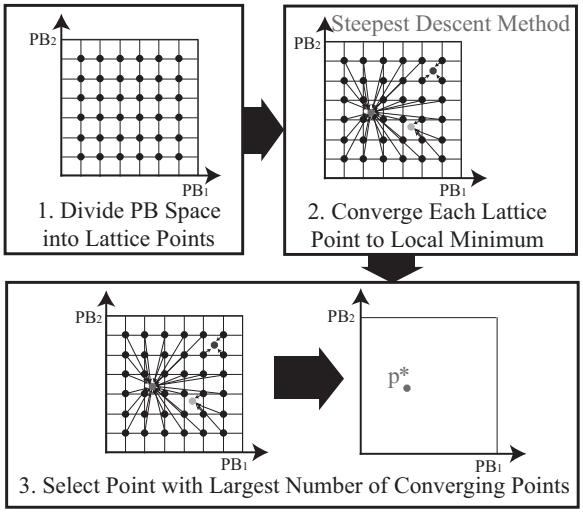


Fig. 5. Overview of consistency evaluation.

tion) and the initial context value is also constant for every pattern. However, the initial object feature value depends on the initial appearance (shape, posture, position, etc.) of the object. Although we use object image as the input for the hierarchical neural network trained in the third phase, the model only contains information about object features at this phase. Therefore, the variety of objects that the model could deal with largely depends on how the object features are set.

The PB to be sought is the one encoding the most reliably predictable 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, the experience of the robot is expected to be represented as the wideness in the PB space, as the update Eqs. (1) and (2) are conducted for every pattern. Under this assumption, the most reliable motion is expected to be the PB with the largest number of points to converge from equally divided initial points (the local minimum PB with the widest converging area). In this method, we divide the PB space defined as $[0,1]$ 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 method is shown in **Fig. 5**.

2.3. Motion Generation Based on Reliable Predictability

The three training phases link the object image to reliably predictable robot/object motion. Using the trained model, motion generation can be conducted as follows:

1. Input object image into hierarchical neural network to calculate PB.
2. Calculate robot motion through *Closed Loop Calculation* by inputting the initial robot motor value, object feature, context value, and calculated PB.

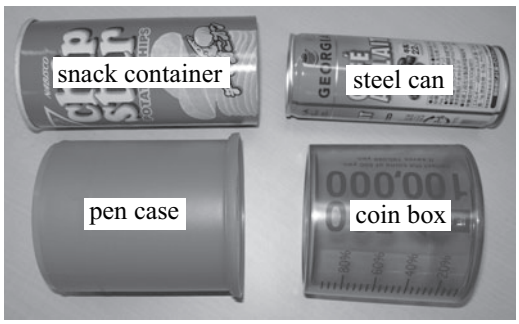


Fig. 6. Cylindrical objects used in experiment.

3. Generate robot motion based on the calculated robot motor sequence.

An important point for calculating robot motion is that the initial robot motor value should be fixed for every training pattern. This is due to the characteristic of *Closed Loop Calculation* that the initial robot motor value is included as the input. Therefore, if a different initial robot motor value is used for training patterns, the model would require a considerably large amount of training data to relate between different initial robot motor values.

3. Setup of Evaluation Experiment

We use the pushing motion of Robovie-IIs with cylindrical objects (**Fig. 6**) to evaluate the effectivity of the method. The snack container and pen case were used for training, while the steel can and coin box were used for evaluation. The snack container and steel can were wrapped with colored paper to ease the process of object extraction. After acquiring motor/image sequences, the image sequences were processed through color thresholding to extract the object regions in the images, and to calculate the object features. We used the center position and the inclination of the longer principal axis of inertia as object features.

The selection of cylindrical objects is due to the following two reasons.

1. Variation of objects depends on definition of object features.
2. Cylindrical objects are easy to evaluate.

Concerning the first reason, as described in the previous section, our method uses initial object feature values for searching the reliably predictable robot motion in the second phase. For example, cylinders and cuboids may have the same object features when facing the same direction, though they have different dynamical properties. The method requires usage of object features that include information about object shape to adapt to different object shapes. One means to deal with this issue is to include an arbitrary variable that discriminates different objects (such as 0.1: cylinders, 0.2: cuboids, etc.). However, this is not practical, and we believe that only through active

sensing can the appropriate object features that also include object shape information be acquired. Autonomous extraction of appropriate object features through active sensing is left as future work. In this paper, as we focus on the effectivity of the motion searching method, we use only cylindrical objects for evaluation. Concerning the second reason, cylindrical objects generate completely different motions when pushed along the longer principal axis of inertia and shorter principal axis of inertia. When pushed along the longer principal axis of inertia, the object would generate unpredictable sliding motions that may also include rotating motions, or an unpredictable rolling motion after the sliding motion. When pushed along the shorter principal axis of inertia, the object would generate predictable, consistent rolling motions in the direction of the shorter principal axis of inertia. Therefore, considering the experiment with cylindrical objects, the objective of the work is to generate pushing motions along the shorter principal axis of inertia of the object to generate rolling motions of the object.

Using these objects, we conducted two experiments. In the first experiment, we considered only laid postures of the objects. In the second experiment, we considered laid and upright postures of the objects. The first experiment is a case where the considered object shape information is included in the object features as only the laid posture is used. The second experiment is a case where the object shape information is not included in the object features as the laid and upright postures have different motion possibilities. In other words, laid cylinders can be rolled or slided, while upright cylinders can be fallen over or slided. Therefore, the first experiment evaluates the effectivity of the method, while the second experiment shows the limitation of the method when appropriate object features are not selected.

3.1. Experiment 1: Evaluation with Laid Cylinders

For the first experiment with laid cylinders, we used 5 planar pushing motions shown in **Fig. 7** created by altering the shoulder roll and elbow yaw axes. Each training object was laid in 5 postures to acquire a total of 50 motion sequences for training. Among the 50 sequences, 33 rolling sequences were acquired, and the other 17 were inconsistent sliding motions. Data were acquired at 2.5 frames/sec for 8 steps. The center position was normalized to $[0,1]$, while the inclination of the longer principal axis of inertia was normalized to $[0.25,0.75]$. Concerning the inclination of the longer principal axis of inertia, defined by $[0,\pi]$, the value inverts during the sequence when it exceeds the limit. This inversion produces discontinuity of the sequence resulting in failure for self-organization of object dynamics during RNNPB training. To prevent this, we manually modified the value when inversion occurred, using the outer limits, $[0,0.25]$ and $[0.75,1]$.

The configuration of neural networks largely affects the training results. In this paper, we empirically decided the number of nodes and parameters of the neural networks.

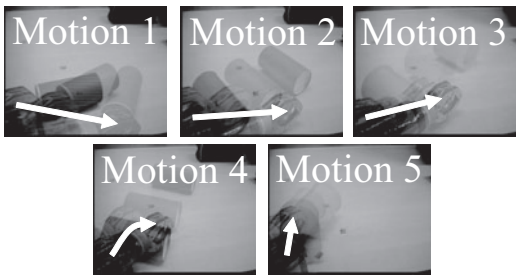


Fig. 7. Robot motions used for training data acquisition.

Table 1. Configuration of RNNPB for experiment 1.

No. of Input/Output Nodes	5
No. of Middle Nodes	15
No. of Context Nodes	15
No. of PB Nodes	2

Table 2. Configuration of hierarchical neural network for experiment 1.

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

The configurations of the neural networks are shown in **Tables 1** and **2**. 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.

3.2. Experiment 2: Evaluation with Different Object Conditions

In this experiment, we consider a case where the cylinders take the laid and upright postures. A laid cylinder would generate rolling or sliding motions when pushed. An upright cylinder would generate falling over or sliding motions when pushed. As we want to evaluate the method with objects containing different motion possibilities, we neglect the sliding motions for the two object conditions. Therefore, for laid cylinders, we generate only rolling motions, and for upright cylinders, we generate only falling over motions.

We created a total of 25 robot motions for acquiring training data. For each of the five motions shown in **Fig. 7**, we also changed the shoulder pitch angle at five levels to generate spatial motions. Neglecting the sliding motions and sequences that the object was occluded, 9 falling over sequences and 52 rolling sequences were acquired. In this experiment, two cameras were used, each calculating the object features (center position and inclination of the longer principal axis of inertia of the object). Data were acquired at 5 frames/sec during 12 steps. The configurations of the neural networks are shown in **Table 3**. We did not use the hierarchical neural network as the motion searching in the second phase does not succeed due to insufficient object features.

Table 3. Configuration of RNNPB for experiment 2.

No. of Input/Output Nodes	9
No. of Middle Nodes	20
No. of Context Nodes	10
No. of PB Nodes	2

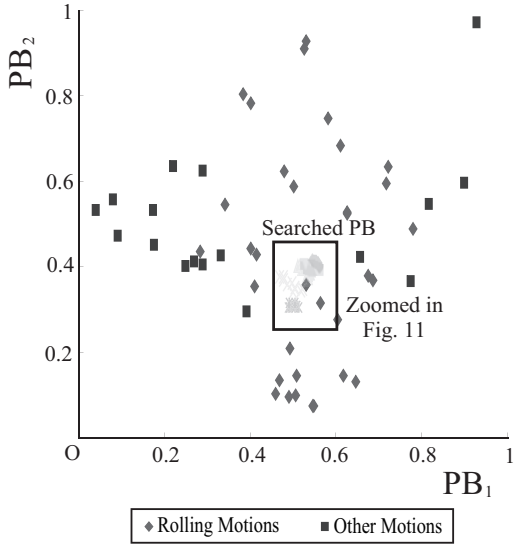


Fig. 8. PB space categorized by object motions.

3.3. Configuration for Motion Searching

For motion searching, we divided the PB space into 10×10 area. 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.

4. Results of Experiment 1: Laid Cylinders

This section describes the results of motion generation experiment with laid cylindrical objects.

4.1. Analysis of the Training Results

By training RNNPB using the 50 training sequences, each sequence is self-organized into the PB, forming the PB space. In this experiment, we analyze the PB space as follows.

1. Categorize PB values based on object motions.
2. Categorize PB values based on robot motions.

These two analysis are the same PB values analyzed from different perspectives.

The first analysis categorizing PB values based on object motions is shown in **Fig. 8**. PB values of rolling motions are shown as rhombi and those of other motions are shown as squares. As can be seen, the PB values of rolling motions are self-organized in the center of the PB space creating a cluster of rolling PB values. **Fig. 9** shows a

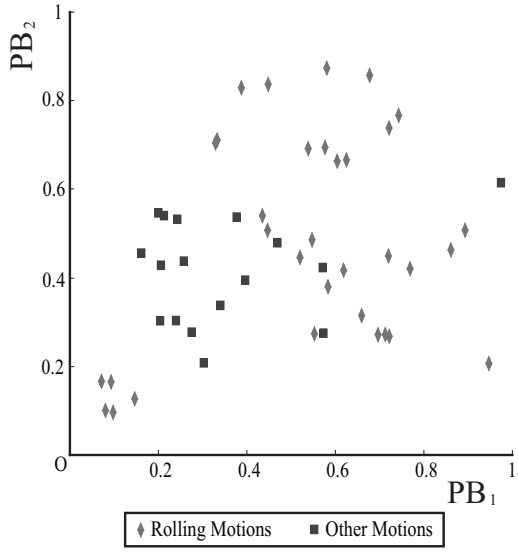


Fig. 9. PB space categorized by object motions without object feature modification.

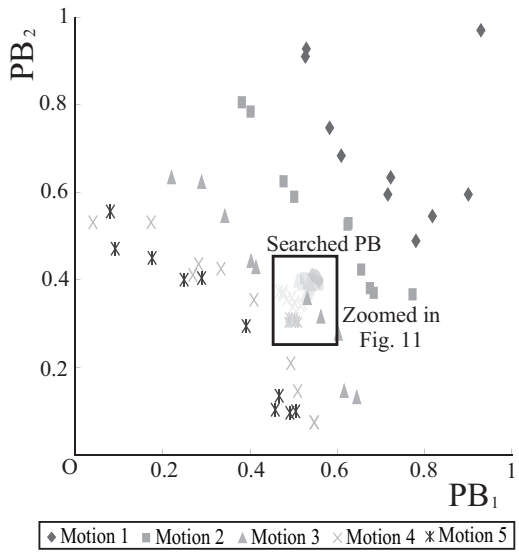


Fig. 10. PB space categorized by robot motions.

training results where the inclination of the longer principal axis inertia was not modified. Some of the sequences in this case contain discontinuities. As the PB space was not properly self-organized in **Fig. 9**, it is notable that the selection of appropriate object features is crucial for training RNNPB.

The second analysis categorizing PB values based on robot motions is shown in **Fig. 10**. The same distribution of PB values are shown in **Fig. 10** as with **Fig. 8**, but with different categorization methods. The PB in **Fig. 10** are categorized based on the robot motions, while those in **Fig. 8** are categorized based on object motions. Motions 1, 2, 3, 4, 5 correspond to the robot motions shown in **Fig. 7**. In other words, Motion 1 is a pushing motion from the left side, and the motion gradually changes to a pushing motion from the front side as changing from Motion 2, 3, 4, 5. From **Fig. 10**, it is notable that as PB_1 and

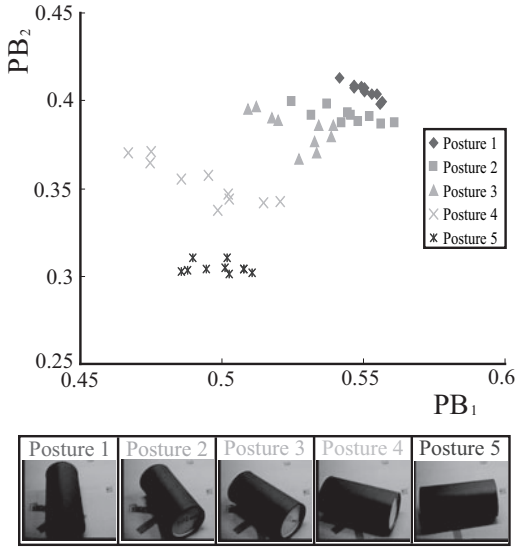


Fig. 11. Zoomed PB space near searched PB.

PB_2 both increase, the robot motion changes to a pushing motion from the left side. On the other hand, as PB_1 and PB_2 both decrease, the robot motion changes to a pushing motion from the front side.

4.2. Analysis of Motion Searching

The PB values searched in the second phase is located in the rectangular area in **Figs. 8** and **10**. The PB values are shown translucently inside the rectangular area. In this paragraph, we present global analysis of the results, while detailed analysis is presented in the next paragraph. The distribution of the PB values possesses two characteristics.

1. The searched PB values exist in the area where the rolling PB are concentrated in **Fig. 8**.
2. The searched PB values exist in the vicinity of the PB values of Motion 3 in **Fig. 10**.

Concerning the first characteristic, the rectangular area representing the distribution of the searched PB values is located near the center of the PB space. As the rolling PB values are distributed in the center area of the PB space in **Fig. 8**, the result implies that the method has searched robot motions that would generate reliably predictable rolling motions for the objects. The second characteristic is a result of the method evaluating reliable predictability from the robot motion perspective. Motion 3 is the most medial motion of the five, that was capable of generating rolling motions for most of the object postures (except when the object was put horizontally). Out of the five motions, the robot has learned that Motion 3 is the most reliably predictable motion. Another interpretation of this result is from the nature of generalization capability. As with most training methods, neural networks are adept at generalizing medial conditions of trained data. Therefore, the model created a more predictable model of the medial motion and this has resulted in distribution of searched PB values near Motion 3.

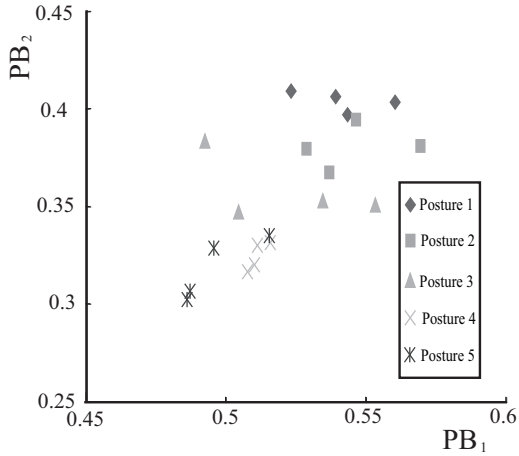


Fig. 12. PB values for motion generation.

A zoom into the searched PB values is shown in **Fig. 11**. The PB values are categorized according to the object postures. We considered 5 postures for training data acquisition (10 data per posture). The initial object feature varies slightly for the data of same postures due to slight changes in object position among the data, since object placement was done by human. The searched PB values are distributed in a *homothetic shape* as the PB distribution of robot motion shown in **Fig. 10**. From **Fig. 11**, it is notable that the PB values form clusters for each object posture. Objects placed in a vertical direction form PB cluster in the top right of the PB space, while objects placed in a horizontal direction form PB cluster in the bottom left of the PB space. Interpreting this result as robot motion, the robot would push objects put in a vertical direction from the left side, and those put in a horizontal direction from the front side. This is a pushing direction along the shorter principal axis of inertia of the object that generates rolling motions. Therefore, from the analysis, it is notable that the robot has learned to generate reliably predictable rolling motions adapting to the posture of the object.

4.3. Motion Generation Experiments

For motion generation experiments, we placed each of the four objects shown in **Fig. 6** in five postures and generated pushing motions for each posture. The PB values calculated for each posture are shown in **Fig. 12**. The PB values are categorized based on the object postures as in **Fig. 11**. The PB values in **Fig. 12** are distributed similarly as the distribution of searched PB values shown in **Fig. 11**. Motion generation results from the calculated PB values are shown in **Fig. 13**. The five arrows in **Fig. 13** show the directions the object has rolled for the five postures. Although the PB values exist in a limited area of the PB space, the robot motions differ greatly for objects placed in Posture 1 and Posture 5. From the experiment, the robot generated pushing motions that induce consistent rolling motions of the cylindrical objects.

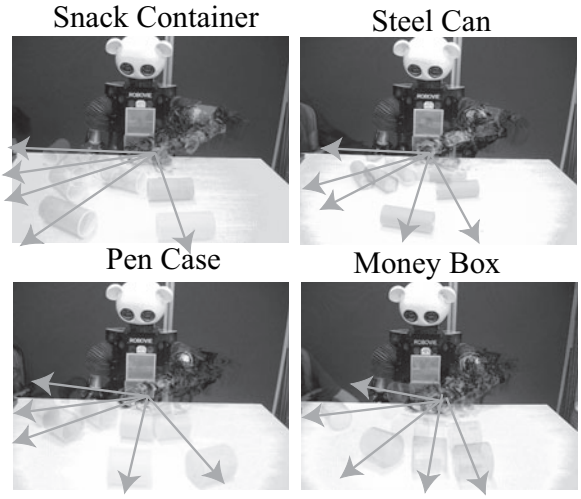


Fig. 13. Motion generation experiments using cylindrical objects.

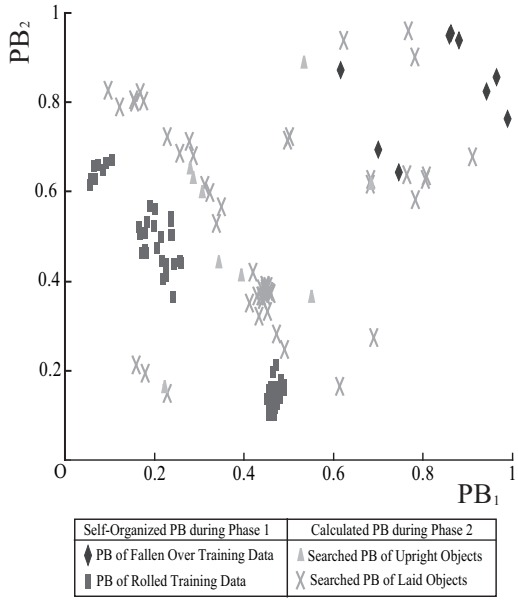


Fig. 14. Self-organized PB space for laid and upright cylinders.

5. Results of Experiment 2: Laid and Upright Cylinders

This section describes the results of motion searching with laid and upright cylinders. The self-organized PB of training data and the searched PB are shown in **Fig. 14**. The rhombi represent PB values of sequences that the object fell over, and the squares represent PB values of sequences that the object rolled. The distributions of these training data are appropriately self-organized based on object motions. The searched PB values of upright objects are shown as triangles, and those of laid objects are shown as "x" marks. From the experimental condition, the PB values of upright objects should be distributed near the cluster of the rhombi, while the PB values of laid objects should be distributed near the cluster of the red squares. This is because upright cylinders can only be fallen over and can't be rolled, whereas laid cylinders can only be rolled and can't be fallen over. However, the results show that the searched PB values are distributed inconsistently.

From the experimental results, RNNPB was capable of self-organizing two object motions (fall over and roll). However, the motion searching results didn't show valid outcomes. This was due to two reasons.

1. Differences in structures of PB space from the first experiment.
2. Insufficiency of object features to discriminate different object conditions.

Concerning the first reason, the PB space in **Fig. 10** had a continuous structure depending on the direction of robot motion. Therefore, it was easy for the method to search through the PB space. However, in **Fig. 14**, the PB space consists of two attractors corresponding to fall over motion and rolling motions. This created a discontinuous space between the two attractors. As our method applies neighborhood searching, the method tended to search for two different motions from slight changes in the initial object features when searching was conducted near the discontinuous space. Concerning the second reason, we used the center position and inclination of the longer principal axis of inertia as object features. Although we used two cameras in this experiment, it was difficult for the model to distinguish the difference between the laid position and upright position from the object features.

6. Discussions

This section presents the discussions considering the experimental results.

6.1. Predefined Object Features

In this paper, we defined the object features as center position and inclination of the longer principal axis of inertia. This predefinition has led to two issues of the method.

1. Requirement of object feature modification to self-organize PB space.
2. Difficulty of distinguishing different initial object conditions.

The first issue is shown in the first experiment. In the experiment, we modified the inclination of the longer principal axis of inertia to prevent discontinuities along the sequence. Using these modified object features, the model was capable of self-organizing the PB space as shown in **Figs. 8** and **10**. The training has created a cluster of rolling motion PB and an axis corresponding to the robot motion. However, without modification, the PB space was not properly self-organized as shown in **Fig. 9**. This was due to the inappropriateness of the inclination of the longer principal axis of inertia to describe the object motion when discontinuity occurs along the sequence. Like so, selection of the appropriate object fea-

tures to describe object motion is a crucial factor for the method.

The second issue is shown in the second experiment. As shown in **Fig. 14**, the model was capable of self-organizing the object motions. However, as the method includes initial object features for motion searching, the model was incapable of distinguishing different object conditions from the object features. Therefore, the searching became sensitive to small changes in initial object features, resulting to inconsistent distribution of the searched PB. To solve this issue, the appropriate object features should be used which also contains information about object shape.

6.2. Scalability for Object Motions

In this paper, we described the experiment using rolling and non-rolling motions of cylindrical objects. In this subsection, we discuss the scalability of the model for adapting to a larger variation of object motion.

In our previous work, we showed the capability of the prediction model using two PB nodes to adapt to four different object motions: slide, fall over, roll, and bounce [8]. Although we trained the model with two motion types (rolling and not rolling) in the experiment in this paper, the model is capable of learning a larger number of motions. Concerning complexity of motions, the bouncing motion considered in [8] is a very dynamic motion. The result in [8] proved that the model is capable of predicting various motion types. The generalization capability of the neural networks have also shown some prediction of a combination of multiple object motions. Thus, the model is capable of dealing with a variety of motions and combination of motions.

Concerning the motion searching method, the model requires some improvement to deal with larger variation of object motions. The current motion searching module inputs object features for predicting object motions, where the prediction model in [8] input object images. Therefore, the model requires object features that represents the object shape in order to adapt to larger varieties of object shapes and motions. By improving the model to self-organize appropriate object features during training, instead of predefining the object features, we believe that the model would be capable of dealing with a larger variation of object shapes and motions.

6.3. Integration with Other Criterion

In this paper, we considered *Reliable Predictability* as the criterion to generate robot motions. The result has shown that the method is capable of generating predictable rolling results for the pushing motion of the robot.

Related works often focus on other criterion to generate motions. Goal-oriented motion generation is one of the major works. These works provide the goal of the object condition (position or posture) and generate the robot motion that would accomplish the goal [6, 7]. Although these works are capable of fulfilling the task, most of these goal conditions can be achieved by several robot motions. Often, some are more reliable than others.

Integration of reliable predictable with goal-oriented conditions for generating motions could be expected to accomplish the tasks more reliably. One means for integration is to perform online searching by introducing the goal-oriented condition as a searching evaluation function in Phase 2. Changing the initial PB for searching would lead to a different local minimum PB. The local minimum PB are candidates of motions that would accomplish the goal. Reliable predictability evaluation could be done to each of the local minimum PB to determine one PB that would most reliably accomplish the task.

7. Conclusions

In this paper, we described a method to generate robot motion based on *Reliable Predictability*. The method consists of a preliminary phase and three training phases. In the preliminary phase, the robot generates active sensing motions to acquire training sequences. These sequences are used in the following three training phases to link object image to reliably predictable robot motion.

1. Train RNNPB using robot/object feature sequences.

2. Search for PB (robot motion) based on reliable predictability evaluation function using steepest descent method.

3. Train hierarchical neural network to link object image with PB calculated in Phase 2.

For evaluation, we used cylindrical objects, due to easiness of evaluation and predefinition of object features. We used modified object features to prevent discontinuity along the sequences. The first experiment using laid cylinders proved that the method is capable of generating reliably predictable rolling motions adapting to the object postures. The analysis has shown that a homothetic distribution of searched PB values was acquired comparing to the PB distribution of training sequences categorized by robot motions. The robot motions tilted to the direction where the objects would be pushed along the shorter principal axis of inertia, inducing the rolling motions of the objects. The second experiment evaluated the limitation of the method using predefined object features. Although the method was capable of self-organizing object dynamics, the searching method failed since it was incapable of distinguishing the differences in object conditions.

As future works, we plan to improve the model to self-organize appropriate object features based on the robot's active sensing experiences. The model would then be capable of generating motions with general objects. Further on, we plan to integrate our model with other motion generation criteria. We believe that these works would contribute to functionalization of affordance to the robot's abilities.

Acknowledgements

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, Grant-in-Aid for JSPS Fellows, RIKEN, and Kayamori Foundation of Informational Science Advancement.

References:

[1] M. Brady, "Artificial Intelligence and Robotics," Artificial Intelligence, Vol.26, pp. 79-121, 1985.
[2] R. Bajcsy, "Active Perception," IEEE Proc., Special issue on Computer Vision, Vol.76, No.8, pp. 996-1005, 1988.
[3] J. J. Gibson, "The Ecological Approach to Visual Perception," Houghton Mifflin, ISBN: 0898599598, 1979.
[4] T. Ogata, H. Ohba, J. Tani, K. Komatani, and H. G. Okuno, "Extracting Multi-Modal Dynamics of Objects using RNNPB," Journal of Robotics and Mechatronics, Special Issue on Human Modeling in Robotics, Vol.17, No.6, pp. 681-688, 2005.
[5] S. Takamuku, Y. Takahashi, and M. Asada, "Lexicon acquisition based on object-oriented behavior learning," Advanced Robotics, Vol.20, No.10, pp. 1127-1145, 2006.
[6] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning About Objects Through Action – Initial Steps Towards Artificial Cognition," Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 3140-3145, 2003.
[7] A. Stoytchev, "Learning the Affordances of Tools Using a Behavior-Grounded Approach," Springer Lecture Notes in Artificial Intelligence, pp. 140-158, 2008.
[8] 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.
[9] J. Hawkins and S. Blakeslee, "On Intelligence," Times Books, ISBN: 0805078533, 2004.
[10] J. Tani and M. Ito, "Self-Organization of Behavioral Primitives as Multiple Attractor Dynamics," IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, Vol.33, No.4, pp. 481-488, 2003.
[11] M. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," Eighth Annual Conf. of the Cognitive Science Society, pp. 513-546, 1986.
[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," in D. Rumelhart and F. McClelland (Ed.), "Parallel Distributed Processing," M.I.T. Press, Vol.1, pp. 318-362, 1986.

Name:
Shun Nishide

Affiliation:
JSPS Research Fellow, Graduate School of Informatics, Kyoto University

Address:
Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan

Brief Biographical History:
2008-2009 Research Fellow (DC2), The Japanese Society for the Promotion of Science (JSPS)
2009- Research Fellow (PD), The Japanese Society for the Promotion of Science (JSPS)

Main Works:
- "Predicting Object Dynamics from Visual Images through Active Sensing Experiences," Advanced Robotics, VSP and Robotics Society of Japan, Vol.22, No.5, pp. 527-546, May 2008.

Membership in Academic Societies:
- IEEE Robotics and Automation Society
- The Robotics Society of Japan (RSJ)
- Information Processing Society of Japan (IPSJ)

Name:
Tetsuya Ogata

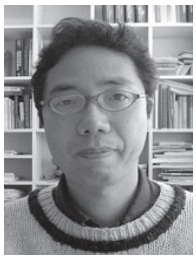
Affiliation:
Associate Professor, Graduate School of Informatics, Kyoto University

Address:
Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan

Brief Biographical History:
1997-1999 Research Fellow, The Japanese Society for the Promotion of Science (JSPS)
1999-2001 Research Associate, Department of Mechanical Engineering, School of Mechanical Engineering, Waseda University, Japan
2001-2003 Research Scientist, Brain Science Institute, RIKEN, Japan
2003-2005 Lecturer, Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Japan
2005- Associate Professor, Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Japan

Main Works:
- "Predicting Object Dynamics from Visual Images through Active Sensing Experiences," Advanced Robotics, VSP and Robotics Society of Japan, Vol.22, No.5, pp. 527-546, May 2008.
- "Experience Based Imitation Using RNNPB," Advanced Robotics, VSP and Robotics Society of Japan, Vol.21, No.12, pp. 1351-1367, Oct. 2007.

Membership in Academic Societies:
- The IEEE Robotics & Automation Society
- The Japan Society of Mechanical Engineering (JSME)
- The Robotics Society of Japan (RSJ)
- Information Processing Society of Japan (IPSJ)
- Japanese Society for Artificial Intelligence (JSAI)
- Human Interface Society (HIS)



Name:
Jun Tani

Affiliation:
Laboratory Head, Brain Science Institute, RIKEN

Address:
2-1 Hirosawa, Wako City, Saitama 351-0198, Japan

Brief Biographical History:
1990-2000 Senior Researcher, Sony Corp.
1995-2002 Visiting associate professor, University of Tokyo
2000-2009 Team Leader, Riken Brain Science Institute

Main Works:
- "Model-based learning for mobile robot navigation from the dynamical systems perspective," IEEE Trans. on Syst. Man Cybern. Part B-Cybernetics, Vol.26, No.3, pp. 421-436, 1996.
- "Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment," PLoS Computational Biology, Vol.4, Issue 11, e1000220, 2008.

Membership in Academic Societies:
- International Society for Adaptive Behavior (ISAB)

Name:
Kazunori Komatani

Affiliation:
Assistant Professor, Graduate School of Informatics, Kyoto University

Address:
Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan

Brief Biographical History:
2002- Assistant Professor, Kyoto University

Main Works:
- "Managing Out-of-Grammar Utterances by Topic Estimation with Domain Extensibility in Multi-Domain Spoken Dialogue Systems," Speech Communication, Vol.50, Issue 10, pp. 863-870, 2008.
- "User Modeling in Spoken Dialogue Systems to Generate Flexible Guidance," User Modeling and User-Adapted Interaction, Vol.15, No.1, pp. 169-183, 2005.

Membership in Academic Societies:
- Information Processing Society of Japan (IPSJ)
- The Institute of Electronics, Information and Communication Engineers (IEICE)
- The Association for Natural Language Processing (NLP)
- Japanese Society for Artificial Intelligence (JSAI)
- The Association for Computational Linguistics (ACL)
- International Speech Communication Association (ISCA)

Name:
Hiroshi G. Okuno

Affiliation:
Professor, Graduate School of Informatics, Kyoto University

Address:
Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan

Brief Biographical History:
1972-1998 Senior Researcher, NTT Basic Research Laboratories
1986-1988 Visiting Scholar, Knowledge Systems Lab, Stanford University
1992-1993 Visiting Associate Professor, Department of Electrical Engineering, The University of Tokyo
1998-1999 Research Manager, JST ERATO Kitano Symbiotic Systems Project
1999-2001 Professor, Department of Information Sciences, Tokyo University of Science
2001-present Professor, Graduate School of Informatics, Kyoto University

Main Works:
- "Computational Auditory Scene Analysis," Lawrence Erlbaum Associates, NJ, 1998.
- "Robust Recognition of Simultaneous Speech By a Mobile Robot," IEEE Transactions on Robotics, Vol.23, No.4, pp. 742-752, 2007.

Membership in Academic Societies:
- The Institute of Electrical and Electronics Engineers (IEEE)
- The Association for Computing Machinery (ACM)
- The Association for Advancement of Artificial Intelligence (AAAI)
- The Acoustic Society of America (ASA)
- The Robotics Society of Japan (RSJ)
- Information Processing Society of Japan (IPSJ)
- Japanese Society for Artificial Intelligence (JSAI)
- Japanese Society for Software Science and Technology (JSSST)
- Japanese Cognitive Science Society (JCSS)