2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Multimodal Integration Learning of Object Manipulation Behaviors using Deep Neural Networks

Kuniaki Noda, Hiroaki Arie, Yuki Suga and Testuya Ogata

Abstract— **This paper presents a novel computational approach for modeling and generating multiple object manipulation behaviors by a humanoid robot. The contribution of this paper is that deep learning methods are applied not only for multimodal sensor fusion but also for sensory-motor coordination. More specifically, a time-delay deep neural network is applied for modeling multiple behavior patterns represented with multi-dimensional visuomotor temporal sequences. By using the efficient training performance of Hessian-free optimization, the proposed mechanism successfully models six different object manipulation behaviors in a single network. The generalization capability of the learning mechanism enables the acquired model to perform the functions of cross-modal memory retrieval and temporal sequence prediction. The experimental results show that the motion patterns for object manipulation behaviors are successfully generated from the corresponding image sequence, and vice versa. Moreover, the temporal sequence prediction enables the robot to interactively switch multiple behaviors in accordance with changes in the displayed objects.**

I. INTRODUCTION

For robots to cooperate with humans in daily living environments, it is essential for the robots to select appropriate behaviors in accordance with environmental changes while coping with the difficulties in handling high-dimensional and large-scale raw sensory inputs. Hence, in most of the robotic applications, the sensory inputs are commonly preprocessed with dedicated feature extraction mechanisms, such as color region extraction or optic flow, for example. These approaches, however, have the side effect that the information filtering by designers possibly neglects essential information and limits the chances for the robot to develop its own capability from the sensory input level.

Meanwhile, deep networks have received increased attention in the machine-learning community and it has been successfully applied to unsupervised feature learning for single modalities such as text [1], image [2], or audio [3]. In such studies, various information signals, even with high-dimensional representation, were effectively abstracted in a reversible manner. The same approach has also been applied to learning fused representation over multiple modalities showing a significant improvement on speech recognition performance [4]. Another study on multimodal integration learning has succeeded in cross-modal memory retrieval by complementing missing modalities [5]. However, most of the current studies on multimodal integration learning utilizing deep networks are focused on finding correlations between static modals such as image and text [6]. Thus, few studies have investigated methods not only for multimodal sensor fusion but also for sensory-motor coordination problems [7] of robot behaviors.

The authors are with Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo, Japan E-mail: kuniaki.noda@akane.waseda.jp

Against these backgrounds, our general interest in this study is to examine the possibility of applying a deep learning framework to the sensory-motor coordination problem on robotic applications, especially with high-dimensional and large-scale raw sensory temporal sequences. To be more specific, the objectives of this paper are (1) to propose a novel approach for application of multimodal temporal sequence learning and cross-modal memory retrieval mechanisms for robot behavior control tasks, and (2) to demonstrate the effectiveness of sensory-motor coordination on object manipulation behaviors by a humanoid robot.

To achieve our objectives, we construct a multimodal integration learning mechanism based on a deep learning framework, and the algorithm called Hessian-free optimization [8] is adopted as a method for training the proposed mechanism. The raw color image inputs acquired from a camera are directly input into an auto-encoder and the corresponding image features are generated. The image features are combined with the joint angles and the multimodal temporal sequence is integrated by another auto-encoder. Our experimental results demonstrate that (1) the proposed method can retrieve temporal sequences over different modalities and predict future sequence from the past, and (2) behavior-dependent unified representations that fuses sensory-motor modalities together are extracted in the temporal sequence feature space.

This paper is organized as follows. In Section II, we briefly review the Hessian-free optimization for training deep networks. In Section III, we describe the general framework of the multimodal temporal sequence learning. In Section IV, we present the practical construction of the proposed framework and the experimental setups for the evaluation. In Section V, we analyze the results, and finally we conclude our work in Section VI.

II. DEEP NEURAL NETWORKS

A. Training deep neural networks

Deep neural network (DNN) is a multilayer neural network model that has more than one layer of hidden units between its inputs and its outputs. Hinton et al. [9] first proposed an unsupervised algorithm to use greedy layer-wise unsupervised pre-training followed by fine-tuning methods for overcoming the higher prevalence of unsatisfactory local optima in the learning objectives of deep models. Subsequently,

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

Martens [8] proposed a novel attempt to introduce a second-order optimization method as the Hessian-free approach for the training of deep networks. The proposed approach effectively and efficiently trained the models by a general optimizer without the pre-training process. We adopt the learning method proposed by Martens for optimizing multiple auto-encoders, one for the self-organization of image features and the other for the temporal sequence learning.

B. Hessian-free optimization

The methodology of the Hessian-free algorithm originates with the conventional numerical optimization theory, known as Newton's method. A canonical second-order optimization scheme such as Newton's method iteratively updates the parameters $\theta \in \mathbb{R}^N$ of an objective function f by computing gradient vector p, and updates θ as $\theta_{n+1} = \theta_n + \alpha p_n$ with a learning parameter α. The primary task of Newton's method is to locally approximate f around each θ, up to the second order, by the quadratic equation,

$$M_{\theta_n}(\theta) \equiv f(\theta_n) + \nabla f(\theta_n)^T p_n + \frac{1}{2} p_n^T B_{\theta_n} p_n, \quad (1)$$

where B_{θ_n} is a damped Hessian matrix of f at θ_n. As H can become indefinite, the Hessian matrix is re-conditioned to be $B_{\theta_n} = H(\theta_n) + \lambda I$, where $\lambda \geq 0$ is a damping parameter and I is the unit matrix

In the standard Newton's method, M_θ is optimized by computing the $N \times N$ matrix B and then solving the system $B_{\theta_n} p_n = -\nabla f(\theta_n)^T$. This computation, however, is very expensive for a large N, which is a common case even with modestly sized neural networks. To overcome this issue, the version of Hessian-free algorithm developed by Martens utilizes the linear conjugate gradient (CG) algorithm for optimizing quadratic objectives in combination with the use of a positive semi-definite Gauss-Newton curvature matrix in place of a possibly indefinite Hessian matrix. The name "Hessian-free" means that the CG does not necessarily require the costly explicit Hessian matrix; instead, the matrix-vector product between the Hessian H or the Gauss-Newton matrix G and the gradient vector p is sufficient. For more details on the concrete implementation, see [8], [10], and [11].

III. MULTIMODAL TEMPORAL SEQUENCE LEARNING MECHANISM

A. Self-organization of image feature vector

High-dimensional image inputs are converted to low-dimensional feature vectors by the auto-encoder (image compression network). The input-output mappings of the image compression network are defined as follows:

$$u_t = f(r_t) \quad (2)$$
$$\hat{r}_t = f^{-1}(u_t), \quad (3)$$

where r_t, u_t, and $\hat{r}_t$ are the vectors representing the input image, the corresponding image feature, and the reconstructed image, respectively. Functions $f(.)$ and $f^{-1}(.)$ represent the transformation mapping from the input layer to the central hidden layer and the central hidden layer to the output layer of the network, respectively.

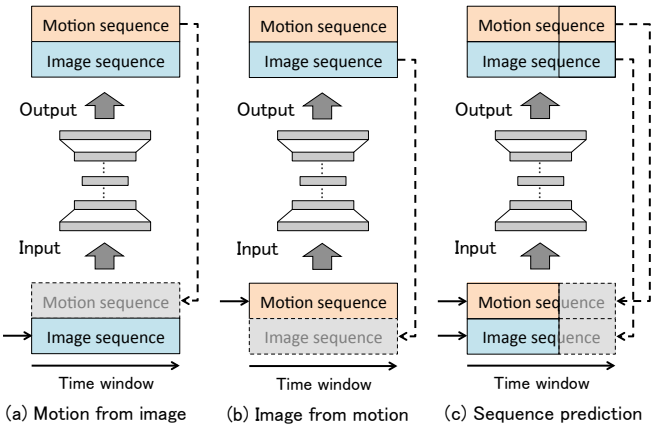

Fig. 1. Cross-modal memory retrieval and temporal sequence prediction

B. Multimodal learning of temporal sequence using time-delay network

A time-delay neural network (TDNN) is a method for utilizing a feed-forward neural network for multi-dimensional temporal sequence learning [12]. Motivated by TDNN, we apply the auto-encoder to the temporal sequence learning problem (temporal sequence learning network). An input to the temporal sequence learning network at a single time step is defined by a time segment of the pair of joint angle vectors and image feature vectors, as follows:

$$s_t = (a_{\mathbf{t}}, u_{\mathbf{t}}) \quad (4)$$
$$\{\mathbf{t} | t - T + 1 \leq \mathbf{t} \leq t\}, \quad (5)$$

where s_t and a_t are the input to the network and the joint angle vector at time t, respectively, and T is the length of the time window. Here, $\mathbf{t}$ represents the previous T steps of the temporal segment from t, and a vector with the subscript $\mathbf{t}$ means a time series of the vector. The input-output mappings of the temporal sequence learning network are defined as follows:

$$v_t = g(s_t) \quad (6)$$
$$\hat{s}_t = g^{-1}(v_t), \quad (7)$$

where v_t and $\hat{s}_t = (\hat{a}_{\mathbf{t}}, \hat{u}_{\mathbf{t}})$ are the multimodal feature vector and a segment of the restored multimodal temporal sequence, respectively. Functions $g(.)$ and $g^{-1}(.)$ represent the transformation mapping from the input layer to the central hidden layer and the central hidden layer to the output layer of the network, respectively.

One of the merits of applying neural networks for multimodal temporal sequence learning is their generalization capability. Because the network can complement deficiencies in the input data, the temporal sequence learning network can be used in two different ways. One way is to retrieve a temporal sequence from one modal for use in another (Fig. 1(a), (b)) and the other way is to predict a future sequence from the past sequence (Fig. 1(c)). Thus, the temporal sequence learning network serves as a cross-modal memory

retriever or a temporal sequence predictor by masking the input data from outside the network in either spacial or temporal ways and iteratively feeding back the generated outputs to the inputs as the substitutions for the masked inputs. The practical implementation of these functions is described in the following subsections.

C. Cross-modal memory retrieval

Cross-modal memory retrieval is realized by providing an input data sequence to either of the two modals and self-generating the corresponding sequence for the other modal inside the network by constructing a recurrent loop from the output nodes to the masked input nodes. The input and output sequences are stored in the buffer (Fig. 1(a), (b)). Hence, in the case of generating a motion from an image sequence, input to the network is defined as follows:

$$s_t = (\hat{a}_{\mathbf{t}}, u_{\mathbf{t}}). \tag{8}$$

In the same way, in the case of retrieving an image sequence from a motion, input to the network is defined as follows:

$$s_t = (a_{\mathbf{t}}, \hat{u}_{\mathbf{t}}). \tag{9}$$

In both cases, the time segment of the recurrent input is generated by shifting the corresponding previous outputs of the network to the time direction for one step by discarding the oldest time step output and filling the latest time step with the value of the newest time step acquired from the output (Fig. 2).

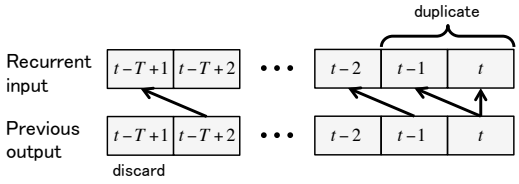

Fig. 2. Buffer shift of the recurrent input

D. Temporal sequence prediction

Similarly, the temporal sequence prediction is realized by constructing a recurrent loop from the output layer to the input layer. The difference is that among all of the T steps of the time window, only the first T_{in} steps (the past T_{in} shifts to the present time step t) of both modals are filled with the input data and the rest (the future $T - T_{in}$ shifts to the predicted time step) are filled with the outputs from the previous time step. Hence, input to the network is defined as follows:

$$s(t) = (a_{\mathbf{t}_1}, \hat{a}_{\mathbf{t}_2}, u_{\mathbf{t}_1}, \hat{u}_{\mathbf{t}_2}), \tag{10}$$

$$\{\mathbf{t}_1 | t - T_{in} + 1 \leq \mathbf{t}_1 \leq t\}, \tag{11}$$

$$\{\mathbf{t}_2 | t + 1 \leq \mathbf{t}_2 \leq t + (T - T_{in})\}. \tag{12}$$

The prediction segment of the recurrent input is generated by shifting the corresponding previous outputs of the network to the time direction for one step (Fig. 3).

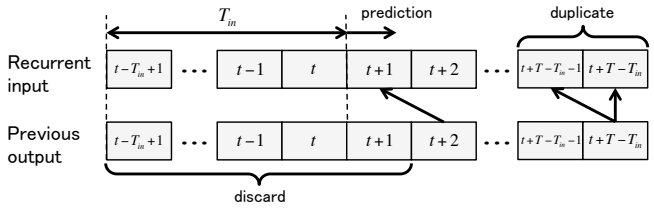

Fig. 3. Buffer shift of the recurrent input for the temporal sequence prediction

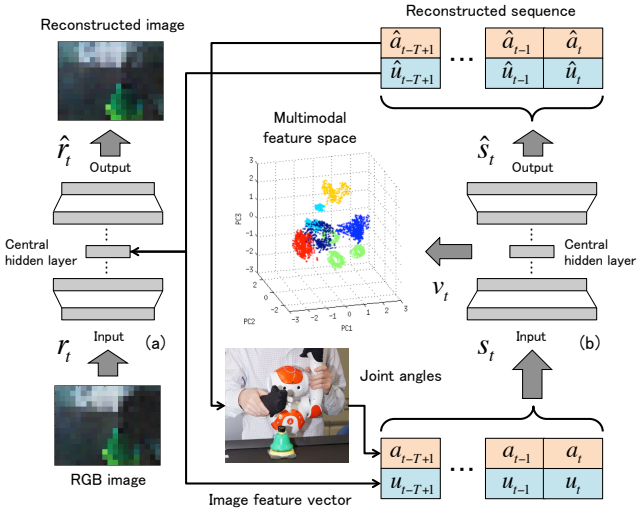

Fig. 4. Multimodal behavior learning and retrieving mechanism

IV. EXPERIMENTS

A. Construction of the proposed framework

Fig. 4 depicts a schematic diagram of the proposed framework. Two independent deep neural networks are utilized for image compression and temporal sequence learning. The image compression network (Fig. 4(a)) inputs raw RGB color images r_t acquired from a camera mounted on the head of the robot and outputs the corresponding feature vectors u_t from the central hidden layer. The image features are synchronized with the joint angle vectors a_t acquired from both arm joints and multimodal temporal segments s_t are generated. The multimodal temporal segments are then fed into the temporal sequence learning network (Fig. 4(b)). Accordingly, multimodal features v_t and reconstructed multimodal temporal segments $\hat{u}_t$ are acquired from the central hidden layer and the output layer of the network, respectively.

The outputs from the temporal sequence learning network is used for both robot motion generation and image retrieval. The joint angle outputs $\hat{a}_t$ from the network are rescaled and sent back to the robot as joint angle commands for generating motion. The network can also reconstruct the retrieved images in the original form $\hat{r}_t$ by decompressing the image feature outputs $\hat{u}_t$ because the image compression network models the identity map from the inputs to the outputs via feature vectors in the central hidden layer.

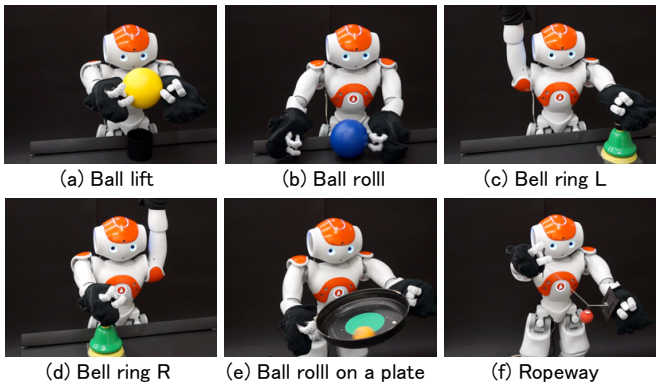

Fig. 5. Object manipulation behaviors. (a) Ball lift: holding a yellow ball on the table with both hands and raising the ball to shoulder height three times with up-and-down movements, (b) Ball roll: iteratively rolling a blue ball on top of the table to the right and left and using both arms alternately, (c), (d) Bell ring L/R: ringing a green bell placed on either the right or left side of the table by the corresponding arm motion, (e) Ball roll on a plate: rolling an orange ball placed in a deeply edged plate attached to both hands and alternately swinging both arms up and down, (f) Ropeway: swinging a red ball hanging from a string attached to both hands by alternately moving both arms up and down.

B. Experimental settings

The performances of the proposed mechanisms are evaluated by conducting object manipulation experiments with the small humanoid robot NAO, developed by Aldebaran Robotics [13]. The multimodal data, including image frames and joint angles, are recorded synchronously at approximately 10 fps. For the image data input, the original 320 $\times$ 240 pixels image is resized to 20 $\times$ 15 pixels. For the joint angle data input, 10 degrees of freedom of the arms (from the shoulders to the wrists) are used.

Six different object manipulation behaviors identified by different colorful toys (Fig. 5) are prepared for training the learning mechanism. We record the multimodal temporal sequence data by generating the different arm motions corresponding to each object manipulation by direct teaching. The resulting lengths of the motion sequence are between 100 and 200 steps (equivalent to between 10 and 20 seconds). To roughly balance the total motion sequence lengths between different behaviors, the direct teaching is repeated 5 to 10 times for each behavior so that the number of repetition becomes inversely proportional to the motion sequence length. Among all the repetitions, one result is used as test data and the others are used as training data. For the multimodal temporal sequence learning, we use a contiguous segment of 30 steps from the original time series as a single input. By sliding the time window by one step, consecutive data segments are generated.

Table I summarizes the datasets and associated experimental parameters. For both the image feature and the temporal sequence learning, the same 12-layered (number of layers of connecting weights) deep neural networks are used. In each case, the decoder architecture is the mirror image of the encoder, yielding a symmetric auto-encoder. The parameter settings of the network structures are empirically determined in reference to the previous works such as [9], or [14]. The

TABLE I

NUMBER OF DATA SAMPLES AND EXPERIMENTAL PARAMETERS

	TRAIN*	TEST*	I/O*	ENCODER DIMS*
Img. feat.	8444	948	900	1000-500-250-150-80-30
Temp. seq.	6848	776	1200	1000-500-250-150-80-30

* TRAIN, TEST, I/O, and ENCODER DIMS give the size of the training data, the test data, the input and output dimensions, and the encoder network architecture, respectively.

input and output dimensions of the two networks are defined as follows: 900 for the image feature learning, which is defined by 20 $\times$ 15 pixels for the RGB colors, and 1200 for the temporal sequence learning, which is defined by the 30-step segment of the 40-dimension multimodal vector composed of 10 joint angles and the 30-dimension image feature vector. For the activation functions, linear functions are used for both of the central hidden layers and logistic functions are used for the rest of the layers.

V. RESULTS

We begin by examining the cross-modal memory retrieval and the temporal sequence prediction performances of the proposed mechanism. Then, we analyze the self-organized structure of the multimodal feature space.

A. Evaluations of cross-modal memory retrieval and temporal sequence prediction

We conduct two experiments for evaluation of the cross-modal memory retrieval performance: one generates the joint angle sequence (motion) by providing image sequences, and the other generates an image sequence by providing the joint angle sequence. For these experiments, inputs for either modal of the full 30 steps are provided and the sequence for the other modal is internally generated in a closed-loop manner (see III-C). In the experiment to evaluate the temporal sequence prediction experiment, the input window length is defined as $T_{in} = 25$ and the corresponding future 5 steps are internally generated as the prediction (see III-D). For all of the experimental settings above, although the initial values for the recurrent inputs are randomly generated, the internal values eventually converge to the corresponding states in association with the input values of the other modal by the generalization capability of the network.

Fig. 6 shows the results of the joint angle sequence generation from the image sequence input and the temporal sequence prediction. The figures on the second row prove that the appropriate trajectories are generated and the configurations of the trajectories are clearly differentiated according to the provided image sequences. The figures on the bottom row show that the proposed mechanism can correctly predict the future joint angles at 5 steps ahead of the 25 steps of the multimodal temporal sequence. The reason for the low reconstruction qualities of the first 30 steps is that random values are supplied for the recurrent inputs at the initial iteration of the generation process. Fig. 7 shows the results of image sequence generation from the joint angle sequence input. In these results, a single frame is drawn among the series of images for each behavior. Although

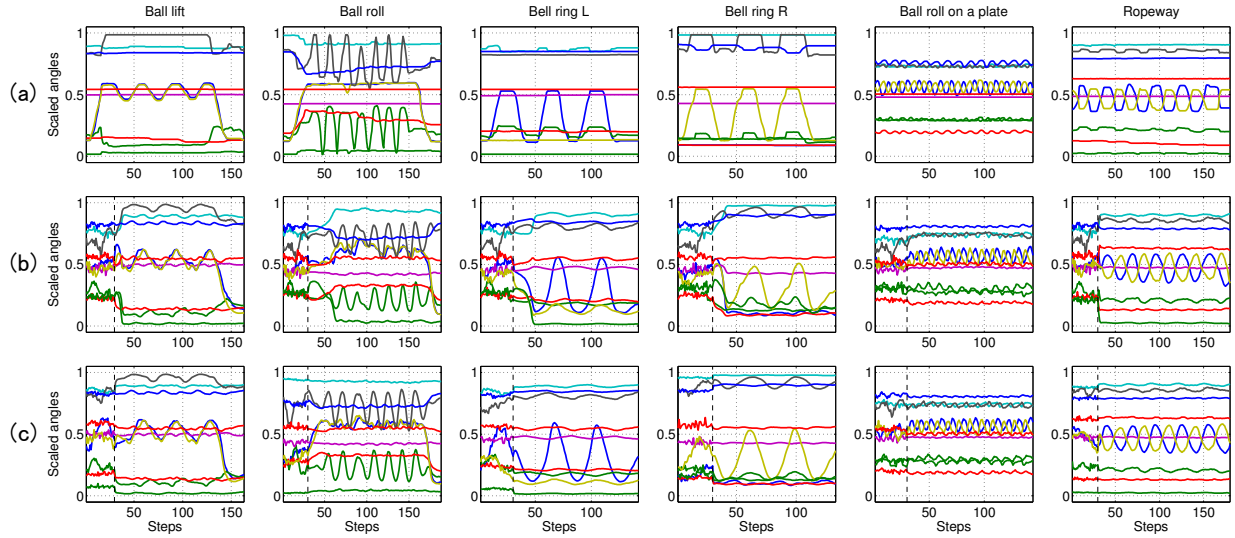

Fig. 6. Motion reconstructions by the proposed model. The graphs on the top row (a) show the original motion trajectories in the test data. The graphs on the second row (b) and the bottom row (c) show the reconstructed trajectories acquired by the cross-modal memory retrieval from the image sequence and the temporal sequence prediction, respectively. The reconstructed trajectories correspond to the same behaviors as the top row. The results from the cross-modal retrieval demonstrate that after the 30th step, it takes about 10 to 30 steps to converge the internal state to produce the correct trajectories.

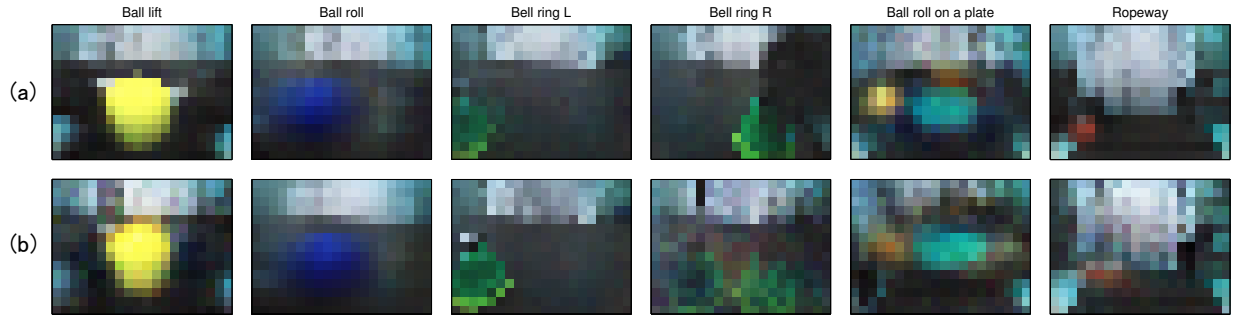

Fig. 7. Image reconstructions by the proposed model. The images on the top row (a) show the original ones decompressed from the image feature vector in the test data. The images on the bottom row (b) show the reconstructed images decompressed from the feature vectors acquired by the cross-modal memory retrieval from the joint angle sequence. The reconstructed images correspond to the same behaviors as the top row.

the details of the images are slightly different, the objects showing up in the images are correctly reconstructed and the location of the color blobs are properly synchronized with the phase of the motion.

Table II summarizes quantitative evaluation results of the cross-modal memory retrieval (IMG $\rightarrow$ MTN: from image to motion and MTN $\rightarrow$ IMG: from motion to image) and the temporal sequence prediction (PRED) performances for the six behavior patterns. The numbers given in each entry of the table represent the root mean square (RMS) errors, range between 0 and 1, of the reconstructed trajectories on the test data.

TABLE II
RECONSTRUCTION ERRORS

	IMG $\rightarrow$ MTN	MTN $\rightarrow$ IMG	PRED
Ball lift	0.0245	0.1440	0.0415
Ball roll	0.0640	0.1012	0.0446
Bell ring L	0.0384	0.0644	0.0235
Bell ring R	0.0274	0.0896	0.0221
Ball roll on a plate	0.0189	0.1349	0.0432
Ropeway	0.0172	0.1150	0.0298

B. Visualization of multimodal feature space

Fig. 8 presents the scatter plot of the three-dimensional principal components of the acquired multimodal features. The multimodal feature vectors are generated by recognizing the training data from the temporal sequence learning network and recording the activations of the central hidden layer. This figure demonstrates that the feature space is segmented according to the different object manipulation behaviors and the feature vectors are self-organizing multiple clusters.

C. Real-time adaptive behavior selection according to environmental changes

As an evolutionary experiment, we try to switch the robot's behavior according to changes in the objects displayed to the robot. The approach is a combination of cross-modal memory retrieval and temporal sequence prediction in the sense that the joint angles 5 steps ahead are predicted from the past 25 steps of the image input sequence. By iteratively sending the predicted joint angles as the target commands

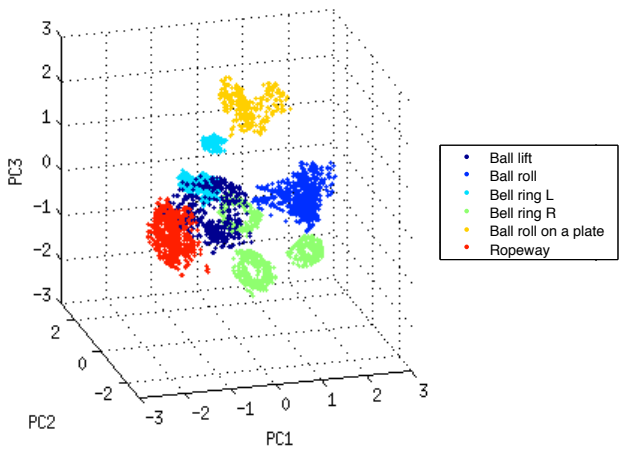

Fig. 8. Acquired multimodal feature space. PC1 to PC3 axes correspond to principal components 1 to 3, respectively.

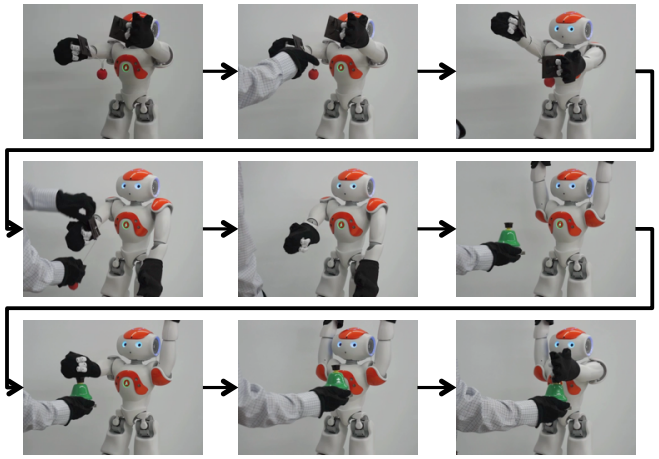

Fig. 9. Real-time transition of object manipulation behaviors according to changes in the displayed objects. The behavior changes are in the order of Ropeway, Bell ring R, and Bell ring L.

for each joint angle of the robot, the robot generates motion in accordance with the environmental changes. For the initial trial, we tested the raw image input and confirmed that the robot can properly generate motion according to changes of the displayed object. However, we found that the reliability for the generated image feature vector is severely affected by the environmental light conditions. Therefore, for the second trial, we went back to the conventional color region segmentation approach and used the coordinates of the center of gravity of the color blobs as a substitution to the image feature vector. With this approach, we successfully enabled the robot to stably switch the appropriate behaviors according to changes of the displayed objects. Fig. 9 shows photos of the transition of multiple behaviors in the real-time and interactive manner.

VI. CONCLUSIONS

In this paper, we introduced a deep neural network for modeling multiple behavior patterns represented by multi-dimensional visuomotor temporal sequences. We presented two applications of the proposed mechanism: the cross-modal memory retriever and the temporal sequence predictor. Their performances were proved by the object manipulation behavior learning experiments conducted in the real-world environment with a humanoid robot. In the experiments, six different object manipulation behaviors were successfully modeled. The analysis of the self-organized feature space revealed that the multimodal features can be utilized as abstracted information for recognizing robot behaviors.

The results from the real-time robot behavior switching experiment revealed that the current approach for utilizing raw image inputs as a means to perceive environment is still not stable enough for handling drastic changes in the environmental lightning conditions. One of the challenges for future study is to improve the robustness of image recognition capability by drawing out the potential of the generalization capability of deep networks by developing methods to train the network with training data sets having more variety.

ACKNOWLEDGMENT

The work has been supported by JST PRESTO "Information Environment and Humans" and MEXT Grant-in-Aid for Scientific Research on Innovative Areas "Constructive Developmental Science" (24119003).

REFERENCES

[1] I. Sutskever, J. Martens, and G. Hinton, "Generating text with recurrent neural networks," *In Proceedings of the 28th International Conference on Machine Learning*, 2011.

[2] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, "Building high-level features using large scale unsupervised learning," *In Proceedings of the 29th International Conference on Machine Learning*, 2012.

[3] G. Hinton, L. Deng, D. Yu, G. Dahl, A.Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[4] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," *In Proceedings of the 28th International Conference on Machine Learning*, 2011.

[5] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," *Advances in Neural Information Processing Systems*, 2012.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, 2012.

[7] J. Dewey, "The reflex arc concept in psychology," *Psychological Review*, vol. 3, pp. 357–370, 1896.

[8] J. Martens, "Deep learning via Hessian-free optimization," *In Proceedings of the 27th International Conference on Machine Learning*, 2010.

[9] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks." *Science*, vol. 313, no. 5786, pp. 504–7, July 2006.

[10] B. Pearlmutter, "Fast exact multiplication by the Hessian," *Neural Computation*, vol. 6, no. 1, pp. 147–160, 1994.

[11] N. N. Schraudolph, "Fast curvature matrix-vector products for second-order gradient descent." *Neural computation*, vol. 14, no. 7, pp. 1723–38, July 2002.

[12] K. Lang, A. Waibel, and G. Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural networks*, vol. 3, pp. 23–43, 1990.

[13] Aldebaran Robotics, "NAO Humanoid," *NAO Datasheet*, Nov. 2012. [Online]. Available: http://www.aldebaran-robotics.com/Downloads/Download-document/192-Datasheet-NAO-Humanoid.html

[14] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for content-based image retrieval," *European Symposium on Artificial Neural Networks*, 2011.