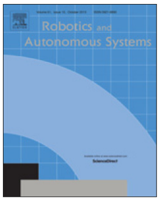
Robotics and Autonomous Systems 86 (2016) 184–195

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



Visual motor integration of robot's drawing behavior using recurrent neural network

Kazuma Sasaki *, Kuniaki Noda, Tetsuya Ogata

Department of Intermedia Art and Science, Graduate School of Fundamental Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan

HIGHLIGHTS

- Bottom-up approach to organize robot's visuomotor experiences.
- Integration learning of drawing behavior by recurrent neural networks.
- Association of drawing motion from drawn picture image.
- Organizing distorted shapes by using drawing experiences.

ARTICLE INFO

Article history:
Received 12 July 2015
Received in revised form 14 May 2016
Accepted 23 August 2016
Available online 13 September 2016

Keywords:
Drawing ability
Drawing robot
Deep learning

ABSTRACT

Drawing is a way of visually expressing our feelings, knowledge, and situation. People draw pictures to share information with other human beings. This study investigates visuomotor memory (VM), which is a reusable memory storing drawing behavioral data. We propose a neural network-based model for acquiring a computational memory that can replicate VM through self-organized learning of a robot's actual drawing experiences. To design the model, we assume that VM has the following two characteristics: (1) it is formed by bottom-up learning and integration of temporal drawn pictures and motion data, and (2) it allows the observers to associate drawing motions from pictures. The proposed model comprises a deep neural network for dimensionally compressing temporal drawn images and a continuous-time recurrent neural network for integration learning of drawing motions and temporal drawn images. Two experiments are conducted on unicursal shape learning to investigate whether the proposed model can learn the function without any shape information for visual processing. Based on the first experiment, the model can learn 15 drawing sequences for three types of pictures, acquiring associative memory for drawing motions through the bottom-up learning process. Thus, it can associate drawing motions from untrained drawn images. In the second experiment, four types of pictures are trained, with four distorted variations per type. In this case, the model can organize the different shapes based on their distortions by utilizing both the image information and the drawing motions, even if visual characteristics are not shared.

1. Introduction

Drawing is an important medium that allows expressing messages or representing one's state of mind through simple lines. In addition, notions depicted in drawing can be shared by others. Investigating the human ability to draw and perceive pictures is significant in understanding human creativity and designing robots that can recognize hand-drawn pictures.

Drawing activities require complex and diverse cognitive skills to perceive visual information, recognize objects or scenes, and generate drawing motion. Therefore, cognitive studies have suggested that the relationship between visual information on drawn pictures and motion is the core component for understanding drawing ability. This relationship is formed in the visuomotor memory (VM), which enables to not only represent pictures by generating drawing motion but also to perceive pictures. Freyd [1,2] noted that humans use information on how the letters are formed, as well as distinctive features or visual characteristics of shape. This suggestion led to subsequent studies, which indicated the effectiveness of temporal order of strokes in letter recognition [3] and the ability to plan a drawing action for representing drawn pictures using the observer's motor system [4]. Recently Waterman et al. [5] suggested a term memory, called

* Corresponding author.
E-mail addresses: ssk.sasaki@suou.waseda.jp (K. Sasaki), kuniaki.noda@akane.waseda.jp (K. Noda), ogata@waseda.jp (T. Ogata).

http://dx.doi.org/10.1016/j.robot.2016.08.022

"visual–motor memory" that is describing the ability to remember visual shapes and use this representation to generate motor activity.

VM is formed from actual drawing experiences through a bottom-up process. Waterman et al. [5] investigated the ability to reproduce simple shapes by drawing after the target picture disappears from the participant's sight. They reported that the accuracy of replicating the target shapes is related to the age of the participants. Pignocchi et al. [4] suggested that visuomotor associations between a drawn line and the required motor activities for drawing it can be learnt from the motor pattern and its perceptual outcome. Furthermore, they indicated that the feature of this visuomotor association is built from learning the interaction between action and perception [6]. Consequently, these studies suggest that VM works as an organized feature of drawing experiences, which comprise motor activities and the temporal transition of drawn pictures corresponding to these activities.

The main purpose of this study is to develop a computational model that can replicate VM from the robot's actual drawing experiences. To build the model, we assume that VM has the following two characteristics:

1. Bottom-up: VM is formed by learning actual drawing experiences.
2. Association: VM allows an observer to associate drawing motion from a drawn picture.

First, the proposed model has to acquire a memory of drawing experiences through a bottom-up process. This means that the model organizes the frames of temporal drawn images and the corresponding motion activities into its memory by learning. In this learning process, the model does not have prior knowledge of the shapes' visual features, such as curvature and edge positioning, or symbolic information on picture classification.

Furthermore, the acquired memory enables robots to associate a drawing motion that reproduces a picture with the picture through a recognition process. This function mimics the above-mentioned motor-perceptual phenomenon of the human's recognition system. In this association process, the acquired memory enables the recovery of a dynamic drawing sequence from a static image, which is already produced. This function is also effective when the observer recognizes distorted shapes. The model distinguishes the different shapes based on their distortions by utilizing both image information and drawing motions, even if visual characteristics are not shared.

Conventional studies on drawing robots have applied a model that can convert captured images into sets of trajectories of the robot's hand. Calinon et al. [7] proposed a humanoid robot system for drawing human portraits. First, this system extracts the main lines of an image using the Canny edge detector [8]. In addition, the model creates a backup of the image in order to maintain the face details, which could disappear after the edge detection. Finally, the model produces the trajectory of the robot's hand, and the robot follows that trajectory. Kudoh et al. [9] explicitly divided the drawing process into developing the shape model of the target object and producing a series of lines that form a hand-drawn-like picture. When the thickness of the lines is not uniform, e.g., when a brush is used, it is necessary to simulate the drawn lines and update the generated trajectory in order to produce a natural order of strokes [10,11,12]. These studies have a common "top-down" approach in the perception of the visual information on the target and the conversion into a trajectory of the robot's end effector by a well-designed strategy. Although this approach contributes to understanding artistic expression by representing drawing styles [13], it is difficult to compare with human's drawing-related cognitive studies, because the characteristic of the model directly depends on the experimenter's policy.

In contrast, a few recent studies have utilized a "bottom-up" approach. Cognitive developmental robotics [14] attempt to develop a corresponding computational model that replicates this set of complex cognitive skills. Mohan et al. [15] proposed a learning model based on the catastrophe theory that uses primitive features of shapes, called "critical points". Their model learns to draw a sequence by decomposing shapes and generates drawing motions by synthesizing the primitive features. Mochizuki et al. [16,17] proposed a dynamical learning model that acquires skills of drawing simple shapes through incremental learning with human–robot interactions. In their study, a robot learns the relationships between the pen's position and the joint angles by moving its right arm randomly. Afterward, the robot develops its drawing skills through incremental learning by adding pause phases at the edge of shapes.

Although the "bottom-up" approach was adopted by some studies, they have not considered both temporal drawn images and motion, which mainly forms the robot's drawing experiences. In order to replicate VM from the robot's actual drawing experiences, a computational model has to integrate these two modalities into a reusable memory not only for presenting a picture by motion but also for associating drawing motion with a drawn picture. Furthermore, this memory allows recognizing distorted variations in drawn pictures, even if they do not share visual characteristics, by using the experiences of drawing motion.

The problem in learning temporal visual features is the large calculation cost for each step of the raw pixel data. To avoid this problem, previous studies utilized common shape features [15] and signal features of picture types for the learning process [18], or replaced the drawn images by trajectories of the end effector of the robots [17].

To overcome the problem of training an image's large dimensionality, we propose here a neural network-based model that can organize temporal drawn images and drawing motions of a robot's unicursal drawing without any handcrafted prepositional knowledge of the picture's visual features, such as curvature and edge detection, or symbolic information on its classification by the visual processing system. Large dimensional temporal drawn images are integrated with drawing motion through multimodal integration learning using deep neural networks (DNNs) [19] and continuous-time recurrent neural networks, which are specified for learning time-series data [20].

The rest of the paper is organized as follows. In Section 2, we introduce the proposed model and describe the functionality acquired by the model. In Section 3, we present two experiments on learning a robot's drawing behavior in order to evaluate whether the proposed model has two VM functions. In the first experiment, the ability to present pictures by generating drawing motions and associate drawing motion with drawn pictures is confirmed. In the second experiment, the possibility to recognize distorted drawn pictures using drawing experiences is evaluated. In Section 4, we discuss the contribution of the present study. Finally, this paper is concluded in Section 5.

2. Computational learning model for self-organizing robot's drawing experiences

2.1. Overview of the proposed model

Fig. 2 presents the architecture of the proposed model. This model is designed to follow the above-mentioned VM characteristics, as shown in Fig. 1.

The first characteristic is achieved by integrating temporal drawn images and motions through self-organized learning (Fig. 1(a)). The original pixel data of the image frames are used as the frame of the drawn picture. The model is trained with this image data and the corresponding motion data, which are the

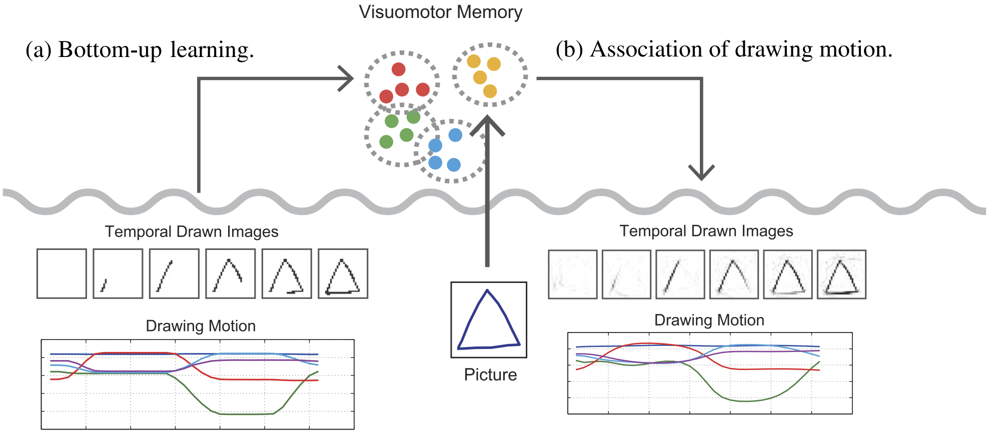


Fig. 1. The overview of the computational model VM acquired through learning robot's drawing experiences.

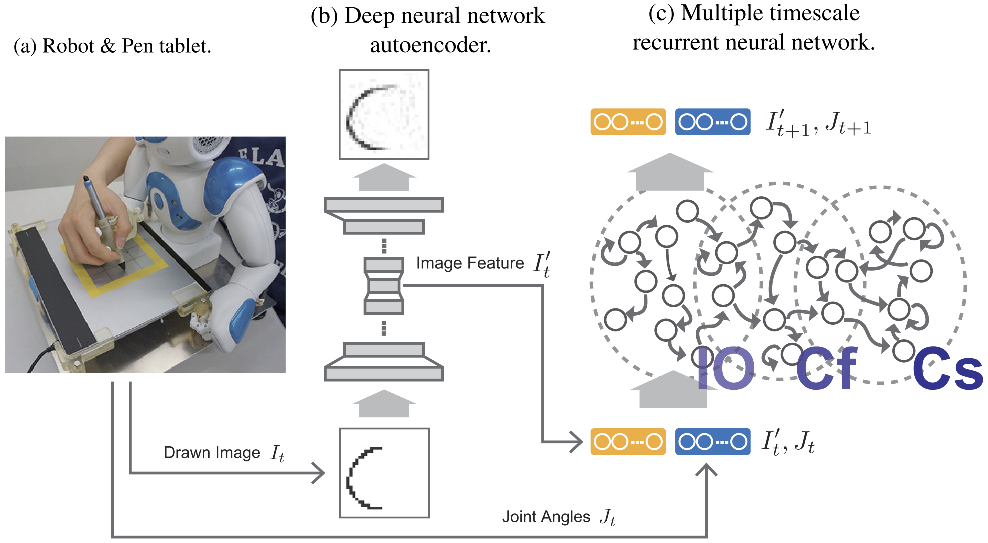


Fig. 2. The overview of the experimental setting and the proposed model for acquiring VM of robot's drawing sequences. (a) A robot that draws pictures on a pen tablet. (b) A deep neural network autoencoder that compresses the drawn images' dimensionality. (c) A multiple-timescales recurrent neural network (MTRNN) for integrating temporal drawn image features by DNN autoencoder and the robot's joint angles.

robot's configuration, i.e., the joint angles. The model is based on multimodal integration learning using a DNN, as proposed by Noda et al. [19]. As mentioned in Section 1, the difficulty of learning temporal drawn images lies in the huge calculation cost when the model needs to process large-dimensional data. The model learns the pixel values of the drawn image at each step. When the model is trained with several sequences, the training dataset becomes too large to converge in real time. Noda et al. proposed a mechanism for integrating multi modal sensory information of robots. In their study, a DNN [21] was applied for compressing the dimensionality of the sensory data by teaching the DNN model to generate identity maps of the input as the output. Afterward, the acquired low dimensionality was used to integrate additional sensory information, e.g., motions.

The second VM characteristic refers to associative functionality. In particular, the model associates drawing motions with images by using the acquired memory through incremental learning (Fig. 1(b)). This function is implemented using continuous-time recurrent neural networks (CTRNNs), which are known as a dynamical system capable of successfully learning temporal sequences [20]. In the proposed model, a CTRNN integrates dimensionally compressed drawn images and motions. Although Noda et al. used a time-delay neural network for multi-dimensional temporal sequence learning, this model is limited in considering the transient sequence's time-scale dependency, because of the time window's length of the input. On the other hand, the CTRNN can learn these types of sequences using back propagation through time (BPTT) [22,23]. For the association process, the CTRNN adapts to minimize the error between the self-generation result and the target picture by re-optimizing the initial state of the context neurons, which determines the network dynamics. This re-optimizing process is also achieved by BPTT, but in this case, only the initial state of the context neurons is updated.

In the present study, a robot assumingly draws line pictures with a pen tablet (Fig. 2(a)). In direct teaching, the drawing sequences are immediately obtained, and include temporal drawn image frames and a time series of the robot's joint angles provided by

$$I = I_0, I_1, \ldots, I_t, \ldots, I_T \quad (1)$$

$$J = J_0, J_1, \ldots, J_t, \ldots, J_T, \quad (2)$$

where I is a temporal drawn image, which consists of a vectorized drawn image frame I_t, J is a robot motion, which consists of a joint angle vector J_t, and T is the length of the sequence. First, the model compresses the dimensionality of drawn images using a DNN autoencoder (Fig. 2(b)). The acquired temporal feature of drawn images and time-series joint angles of the robot are learnt by CTRNN.

2.2. DNN

DNN is a feed-forward neural network model comprising multiple fully connected layers. Hinton et al. showed that activated values in the central DNN layer represent dimensionally compressed input features when the network is trained to encode the input as the output [21]. The nth middle layer's output $\boldsymbol{\epsilon}_n$ is computed as follows:

$$\boldsymbol{\epsilon}_n = sigmoid(W_{n-1}\boldsymbol{\epsilon}_{n-1} + \boldsymbol{\beta}_{n-1}), \tag{3}$$

where W is the weight matrix and β is the bias vector.

To train the network, a truncated Newton-optimization method is applied, called Hessian-free optimization [24]. This method is based on the standard Newton method and computes the gradient vector p to update the network's parameter θ as $\theta_{n+1} = \theta_n + \alpha p_n$ with learning parameter α described as follows:

$$M_{\theta_n}(\theta) = f(\theta_n) + \nabla f(\theta_n)^{\mathrm{T}} p_n + \frac{1}{2} p_n^{\mathrm{T}} B_{\theta_n} p_n, \tag{4}$$

where ∇f is the gradient of the cost function f and B is a damped Hessian matrix of f. In the Hessian-free approach, a positive semi-definite Gauss–Newton curvature matrix, obtained in the linear conjugate gradient for $M_{\theta_n}(\theta)$, is used instead of matrix B, which is extremely expensive for a large network [25]. By applying the Hessian-free optimization, training DNNs avoid several unsatisfactory local optima.

2.3. CTRNN

Multi timescale recurrent neural network (MTRNN) [26], which is a type of CTRNN, is applied to integrate temporal drawn images and drawing motions. CTRNN's neurons activities are decided not only by synaptic inputs but also by the history of the neural state. The firing rate of neuron $\dot{u}_{i,t}$, which has the time constant τ_i, is described as follows:

$$\tau_i \dot{u}_{i,t} = -u_{i,t} + \sum_j w_{ij} x_{j,t}, \tag{5}$$

where $u_{i,t}$ is the internal state of ith neuron in tth step and w_{ij} is the weight from the activation of jth neuron $x_{j,t}$ to ith neuron. In MTRNN, the model includes input–output neurons (IO unit) and no-input–output neurons (context unit). Through the IO unit, MTRNN predicts the next step of input data as its output. Neurons in the IO unit connect with the context unit through recurrent connections. Furthermore, the context unit is divided into the fast context unit (Cf unit) and the slow context unit (Cs unit) by the difference of its recurrent connections and the time constant values τ, which correspond to the speed of the changing neuron's dynamics. The Cf unit has lower time constant values than the Cs unit. In addition, the Cf unit has recurrent connections not only with the IO unit but also with the Cs unit. In contrast, the Cs unit exhibits higher constant values and connects only with the Cf unit.

In the actual forward propagation, the internal states are computed by the following equation:

$$u_{i,t} = \begin{cases} \left(1 - \dfrac{1}{\tau_i}\right) u_{i,t-1} + \dfrac{1}{\tau_i}\left(\displaystyle\sum_{j \in N} w_{ij} x_{j,t-1}\right) & (t \neq 0) \\ 0 \quad (t = 0 \wedge i \in IO, Cf) \\ Cs_{t=0} \quad (t = 0 \wedge i \in Cs) \end{cases} \tag{6}$$

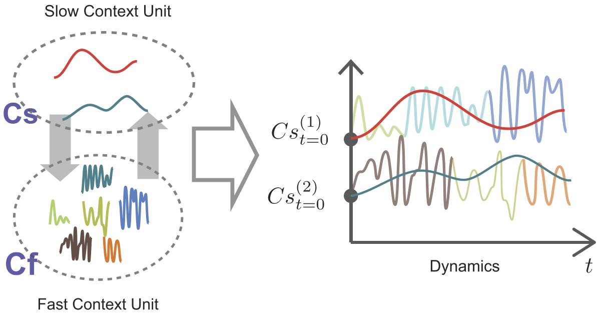


Fig. 3. An example of MTRNN's dynamics. In this figure there are two sequences, which have different initial values than the Cs unit Cs.

where Cs_0 is the initial context value in the Cs unit's neurons. Each neuron is activated by the following sigmoid function:

$$x_{i,t} = sigmoid(u_{i,t}). \tag{7}$$

After propagating for a single step, the activation values in the IO unit x_t are copied as next input:

$$x_{i,t+1} = \begin{cases} x_{i,t} & (i \in IO \wedge t \neq 0) \\ \hat{x}_{i,t} & (i \in IO \wedge t = 0) \end{cases} \tag{8}$$

where $\hat{x}$ is the target sequence for training the network.

MTRNN is trained by BPTT [22], which is a general optimization method for recurrent neural networks. The loss L_t between the generated sequences by MTRNN and the target sequences is defined as the mean square error as follows:

$$L_t = \frac{1}{2} \sum_{i \in IO} (\hat{x}_{i,t} - x_{i,t})^2. \tag{9}$$

The batch-wise training method is used for all the training sequences s to update the network's parameters θ, as follows:

$$\theta_{n+1} = \theta_n - \alpha \frac{\partial L(\theta_n)}{\partial \theta_n} \tag{10}$$

$$\frac{\partial L(\theta_n)}{\partial \theta_n} = \sum_s \sum_t \frac{\partial L(\theta_n)^{(s)}_t}{\partial \theta^{(s)}_{n,t}} \tag{11}$$

where n is the number of iterations and α is the learning rate.

When MTRNN generates temporal sequences by assigning the next step's input from the output, it reconstructs learnt dynamics from the combination of two types of temporal values in the context unit (Fig. 3). Because of this multi-timescale capacity and the hierarchical connectivity between the context units, MTRNN can effectively learn complex sequences and organize them as a combination of two context units. In particular, the initial Cs unit context values $Cs_{t=0}$ are trained as the organized low-dimensional feature, which controls the entire dynamics of MTRNN's behavior when the network generates a sequence. The initial Cs unit value strongly affects the generated sequence because MTRNN's dynamics depend on the differences between the initial input from the IO unit and the Cs unit value (Cf unit's initial values are always zero).

In the case of learning robot's drawing behavior, the initial Cs unit value will correspond to the trained drawing sequences. Each drawing sequence has both the temporal drawn image and the time-series joint angle of the robot. By the training process, these two dynamics will be merged into a trajectory in MTRNN's space. Each trajectory has each own Cs initial value at the first step in the generation process, i.e., drawing behavior.

2.4. Learning drawing sequences

Although MTRNN can develop a self-organized memory from time-series data, there are limitations in optimizing large dimensional input, such as drawn image pixels. The imbalance between the number of dimensions in the images and in the joint angles creates an error, which is regressing toward the large dimensional dataset, i.e., the images. In addition, MTRNN incurs a substantial calculation cost because the number of network parameters corresponds to the input data and the context unit dimensions.

Therefore, the vector (I_t, J_t) is replaced with (I'_t, J_t) as input data in MTRNN, where I'_t is the activated value vector of the central hidden DNN layer. For both MTRNN and the DNN (DNN–RNN) training process, DNN is initially trained to reconstruct I_t as the output $\hat{I}_t$:

$$I'_t = dnn(I_t) \tag{12}$$

$$\hat{I}_t = dnn^{-1}(I'_t) \tag{13}$$

where *dnn* is the forward propagation from the input layer to the central hidden layer and dnn^{-1} refers to the opposite process from the central hidden layer to the output layer. Following this, MTRNN processes the training drawing sequences, which are composed of the temporally compressed image feature I' and the time-series joint angles J. Finally, MTRNN integrates these two types of sensory inputs in the self-organized memory, composed of the initial context value of the Cs unit at the beginning of its generation process.

To conduct the training of the proposed model, the training dataset is collected by teaching the robot to draw the training pictures. When the robot moves followed by the prepared motion by the experimenter, all frames of the drawn image and corresponding joint angles are recorded. In the present study, the drawn image is captured using a pen tablet. In the experiments, we rendered the drawn image from captured the pen's position. Then, the proposed model can be trained by both the rendered image frames and joint angles.

2.5. Functionality of the proposed model

After the training process, the DNN–RNN model can generate the trained drawing sequences by using the parameters acquired by the training. At the beginning of the generation process, one of the initial values is assigned to the MTRNN's Cs unit. Then, the MTRNN recursively predicts the next step of the drawn image feature and the corresponding joint angles by receiving the current sensory information of the robot and drawn picture. At each step, the captured drawn image is dimensionally compressed by the trained DNN. After that, the compressed image feature and the joint angles read from the robot's sensors are input to the MTRNN. Further, the joint angles of the MTRNN's output become the motor command to the robot.

Further, the DNN–RNN model can associate sequences from a drawn image using the learnt parameters. This association process is realized by applying BPTT (Fig. 4). First, the dimensionally compressed image features of the target image and the white image are computed by the trained DNN. Following this, MTRNN generates a drawing sequence by using the image feature of the white image and the given joint angles as the initial input, with the initial Cs unit values. After generating for arbitrary steps, the error between the generated image feature and the target image feature is computed. This error is used to optimize the initial Cs unit values using BPTT and the corrected weights of MTRNN. Therefore, the retrained initial Cs unit value becomes an appropriate value for generating drawing motion from the given target image.

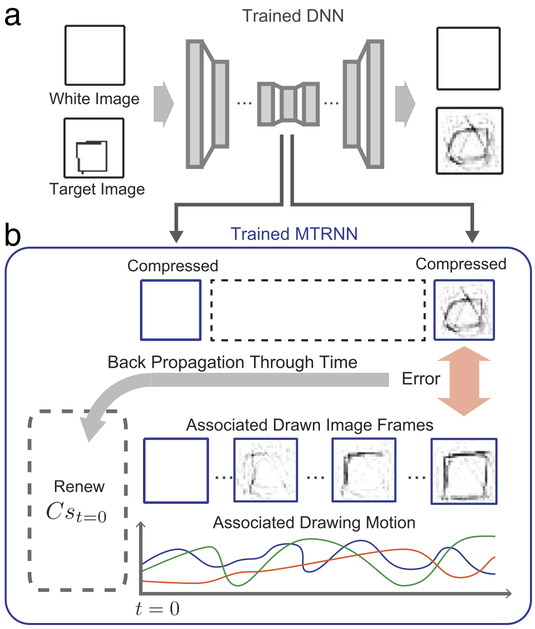


Fig. 4. Overview of the process for obtaining the re-optimized initial Cs values. (a) Acquiring dimensionally compressed drawn image features of the white image and the target image. (b) BPTT optimization using the error between the target image's feature and the generated image's feature by MTRNN with temporary initial values of the Cs unit.

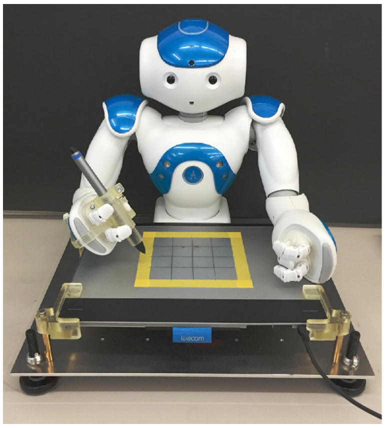


Fig. 5. A robot draws on a pen tablet. The pen tablet is fixed with the robot's right hand by an adapter.

3. Experiments on learning robot's drawing behavior

3.1. Experiments on association of drawing motion from drawn image

Experimental setup

The proposed model is tested by experiments on the association of drawing motions using the small humanoid robot NAO, developed by Aldebaran Robotics [27]. This robot draws drawing sequences to prepare the training dataset by direct teaching. An Intuos-pen tablet [28] is used to capture the drawn images at each step. The robot and the pen tablet are fixed on a base-plate to avoid capturing errors as shown in Fig. 5. In addition, the pen is placed in the robot's right hand with an adapter, which allows the pen to move vertically, again to avoid capturing errors imparted by the pen tip lifting from the tablet.

The training data includes 15 drawing sequences for direct teaching. These sequences contain three types of shapes: circles,

Table 1
Number of training picture images and experimental parameters. IO, DIMS, DATA, TRANS, ROTATED, and Training Iter give the dimension of the IO neurons, the network's dimensional structure, the number of the recorded training data, the translated training data, the rotated training data, and the optimization iteration, respectively.

	IO	DIMS	DATA	TRANS	ROTATED	Training iter
DNN	900	900-400-180-80-30-10-30-80-190-400-900	494	3,1940	2910	100
MTRNN	15($\tau = 1$)	Cf($\tau = 12$): 30, Cs($\tau = 60$): 20	494	–	–	15,000

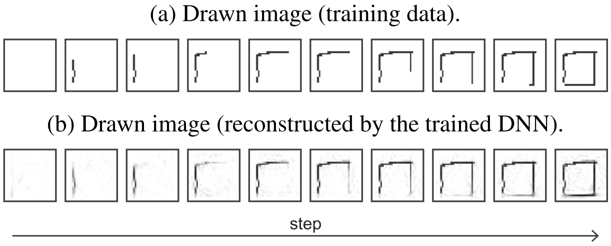

(a) Drawn image (training data).
(b) Drawn image (reconstructed by the trained DNN).
step

Fig. 6. An example of the reconstructed temporal drawn images; (a) the original images of the training dataset. (b) The reconstructed images by the trained DNN.

triangles, and squares; each type is drawn five times. The five variations for each shape have approximately the same initial point, and all pictures are drawn clockwise with one stroke. In addition, 30 × 30 pixel black-and-white drawn image frames are captured. The length of the recorded drawing sequences is between 25 and 50 steps (equivalent to between five and ten seconds). Together with capturing temporal drawn images, time-series of five angles corresponding to the DoF of the robot's right arm are also obtained.

The structure of the DNN–RNN model is decided empirically, but the hyperparameters of the networks are determined a priori. First, the layer structure of DNN is designed to compress the image dimensionally into a low-dimensional vector, which can be trained with joint angles. The size of the compressed image feature vector should be close to the number of joint angles, because the MTRNN training will suffer from the size imbalance of the error, which directly concerns to the training performance of BPTT. In terms of the number of hidden DNN layers, we followed the structure of the model proposed by Noda et al. The number of neurons in the MTRNN's Cf and Cs units is decided after trying several combinations of the parameters. We choose one of the parameter sets, which can minimize error after the convergence. The value of Cf and Cs units strongly depends on the drawing speed taken by direct teaching. Therefore, we utilize another set of parameters between the two experiments.

Table 1 presents the experimental settings of the DNN–RNN model. First, the DNN processes 100 iterations using the Hessian-free optimization method to acquire 10 dimensionally compressed image features of vectorized 30-by-30 pixel images. The training dataset includes not only the drawn image frames but also the same translated and rotated images in order to have a wider spatial variability range for these shapes. After training DNN, the MTRNN processes 15,000 iterations using BPTT with the training dataset, which has 15 dimensions, including 10 dimensions of the temporal image features and the joint angles of the robot.

Generating the training sequences

After the training process, the ability to memorize the training sequences is confirmed by generating with the trained parameters. First, the generated sequences are controlled by the trained model using $(I_{t=0}, J'_{t=0})$ for the initial input, and $Cs_{t=0}$ for the initial values of the Cs unit. Fig. 6 shows an example of the generated drawn image sequences by the trained model. To depict these images, the trained MTRNN temporal image features are reconstructed by the latter part of the trained DNN's forward propagation dnn^{-1}. The generated temporal drawn images maintain the visual information. In addition, Fig. 7 depicts examples of the motion trajectories and the drawn pictures, which are obtained by the generated sequences. Note that (b) is a reconstruction of the drawn image (a) by DNN. The drawn lines in (c) are colored according to the calculated speeds of the pen tip (d), when the robot moves with the generated motion, as shown in (e). The drawn lines maintain the shape characteristics of the training drawn images, and the model reconstructs these training images. Consequently, the proposed model has the capacity to memorize the dynamics of the training drawing sequences through a bottom-up learning process.

Association of the drawing motions with the drawn images

The DNN–RNN model associates the drawing motions by retraining the initial Cs unit values as described by 2.5. The association is configured for 45 steps, and the pen tip position is set at the left bottom side of the canvas. Fig. 8 summarizes the association results for the non-trained images in the same manner as Fig. 7. The reconstructed results of the temporal drawn images (b) indicate that the circle and the triangle are clearly associated, in contrast with the results for the square, which are distorted at the edge points. In addition, the initial point and the ending point do not match in all three cases.

These distortions in the associated drawn lines are attributed to the characteristics of the CTRNN. The training drawing motion's trajectories change in a discontinuous manner at the edge points. The CTRNN cannot completely recover the edge points, because they are approximated by drastically changing but continuous dynamics. This CTRNN characteristic also causes the mismatch between the initial and the ending point. At the beginning part of the drawing motion, the pen's position springs back in the direction opposite to the direction the pen has to move, because the initial startup of the joint angles needs the CTRNN to generate discontinuous trajectories.

Although the drawn lines include distortions, the associated motions maintain dynamical characteristics as shown in (c), (d), and (e) of Fig. 8. The speed of the pen tip decreases at the corners of the drawn triangle and square. At the end of the circle's associated motion, the speed drastically decreases at the right side of the canvas because the length of the memorized circle's drawing sequence is shorter than that of the other shapes. Consequently, the proposed model can associate drawing motions for the three types of pictures from non-trained similar pictures using the learnt VM.

Visualization of the acquired feature by the model

Fig. 9 represents the three-dimensional principal components of the acquired temporal drawn images. These features form linear shapes, which correspond to a drawing sequence. Each temporal feature shares the same value at the begging of sequences, i.e., the white image. These temporal features are discriminated by the spatial distribution of the black pixels.

Fig. 10 presents the three-dimensional principal components of the temporal Cs unit value. The distribution of these features resembles that of the feature acquired by DNN. The time-dependency of the networks creates a difference between these two features at the beginning point. DNN cannot discriminate dynamic sequences

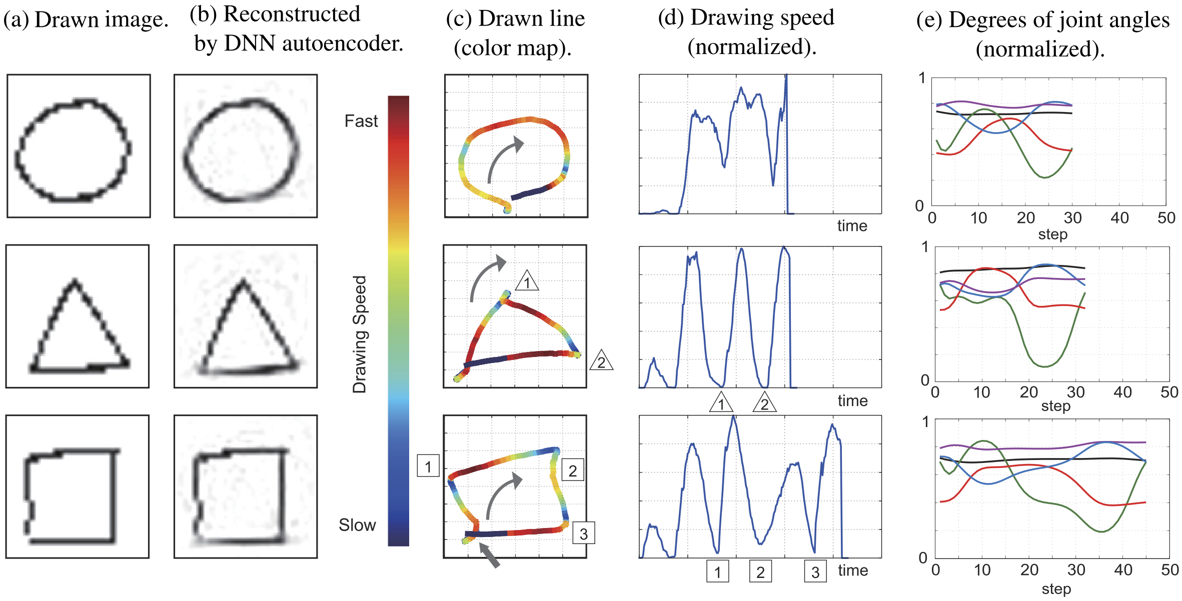


Fig. 7. The generation of the training dataset. (a) Drawn images at the end of the sequences. (b) Reconstructed images by the trained DNN. (c) Lines drawn through generated motion by the robot. (d) Speed of the pen's position, with numbers corresponding to the corners of the respective drawn lines. (e) Generated time-series of joint angles corresponding to each DoF of the robot's arm. Note that the value of (d) and (e) is normalized by the maximum value in order to make the figures more readable.

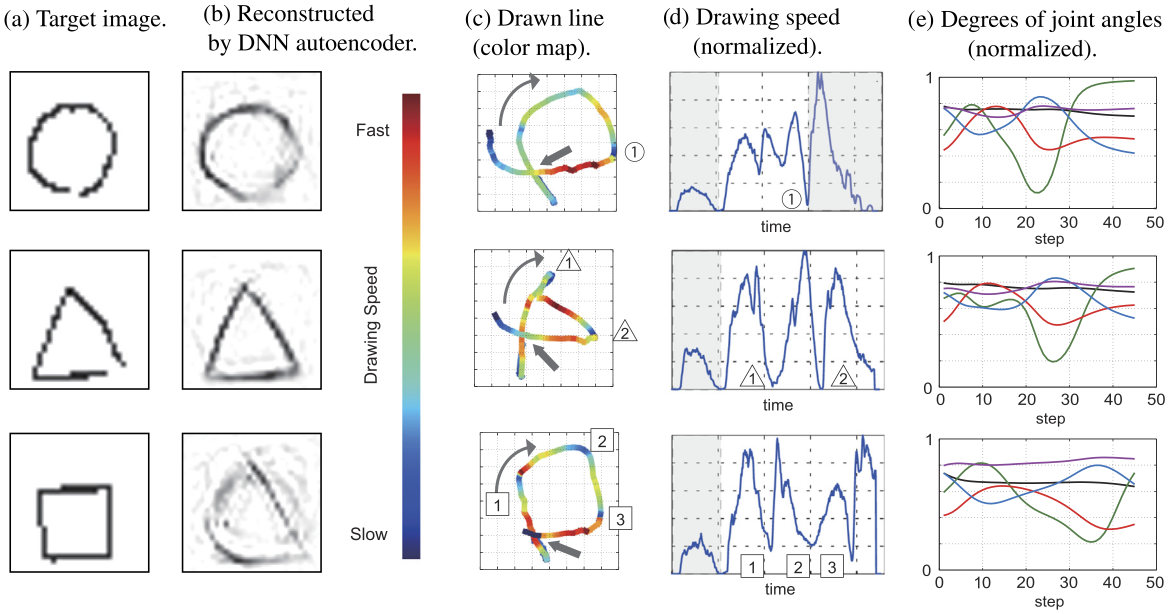


Fig. 8. Association results of the non-trained picture images. (a) Target images to associate the drawing sequence; (b) A reconstructed final drawn image frame; (c) Drawn lines by the robot and the lines colored according to the speed of the pen's position; (d) Speed of the pen's position, with the numbers corresponding to the corners of the respective drawn lines; (e) Time-series joint angles associated by MTRNN. Note that the value of (d) and (e) is normalized by the maximum value in order to make the figures more readable.

in the initial step because it cannot process temporal relationships in its structure or through the learning algorithm. In contrast, MTRNN's behavior strongly depends on the initial Cs unit values, which are learnt by BPTT. These values affect the generation sequences continuously through the recurrent process.

3.2. Experiments on learning distorted shapes

In the previous experiments, the proposed learning model successfully memorized the drawing sequences, which comprised three types of shapes, and associated drawing motion with the non-trained drawn images using acquired memory. In this section, the possibility to discriminate drawn "sloppy" shapes by visuomotor experiences is investigated by experimenting with the same setup conditions (robot and pen-tablet).

Experiment setup

Fig. 11 presents the pictures drawn by the robot with direct teaching. The prepared 16 drawing sequences consist of four types of shapes: circles, hearts, moons, and triangles. The shapes are clockwise drawn with a single stroke, beginning from the almost same point for each shape. The shape lengths range from 30 to 40

Table 2
Number of training picture images and experimental parameters. The IO, DIMS, DATA, TRANS, ROTATED, and Training Iter give the dimension of IO neurons, the dimensional structure of the networks, the number of the recorded training data, the translated training data, the rotated training data, and the iteration for the optimization, respectively.

	IO	DIMS	DAT	TRANS	ROTATED	Training iter
DNN	900	900-400-180-80-30-8-30-80-190-400-900	631	40,384	3786	100
MTRNN	13($\tau = 1$)	Cf($\tau = 3$): 30, Cs($\tau = 30$): 5	631	–	–	15,000

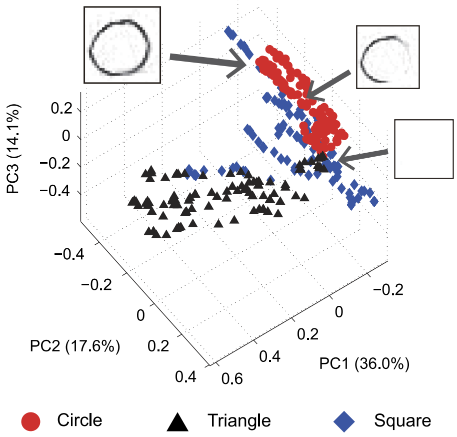


Fig. 9. Acquired features of drawn picture image frames (Training data set). PC1–PC3 axes correspond to principal components 1–3 with contribution values, respectively.

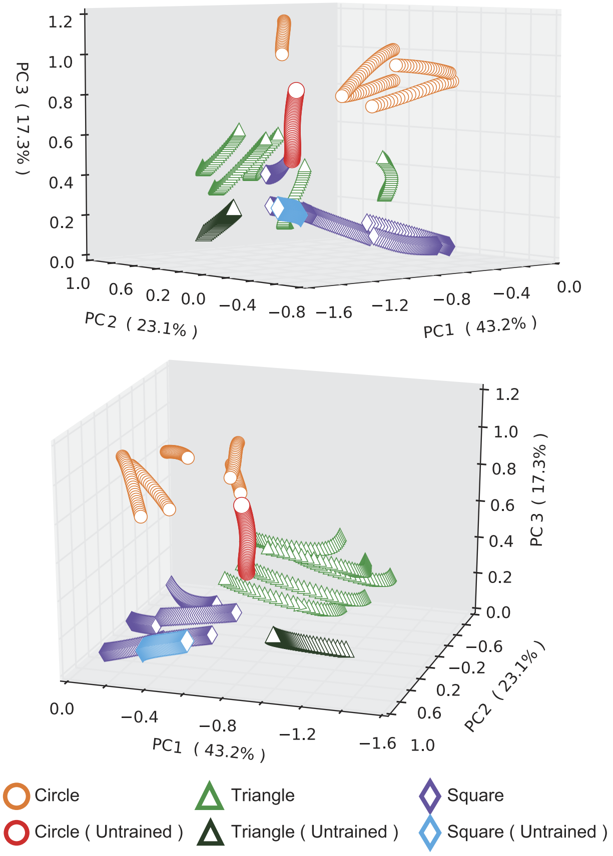


Fig. 10. Acquired features of the visuomotor experiences in slow-context units of the MTRNN in the training and the associated sequences. These two figures correspond to the same 3D-plots viewed from different angles. The PC1–PC3 axes correspond to the principal components 1–3 with their respective contribution values.

steps, and each black-and-white image frame has 30-by-30 pixels. As in the previous experiments, five joint angles were obtained from DoF of the robot's right arm. The experimental parameters of the proposed learning model are listed in Table 2.

The training dataset includes four variations for each shape, divided according to the degree of distortion. Vertically deformed shapes are broad ones, and horizontally deformed shapes are higher than vertically deformed ones. In addition, two types of temporally deformed shapes are prepared: overdrawing and uncompleted shapes. Although the visual characteristics differ from each other, the captured drawing motions share the same features as the changing pattern depicted in Fig. 12.

Comparison of shape features

To verify the present hypothesis, the ability to recognize distorted shapes under different model conditions is evaluated. In particular, the distribution of the drawn image is compared, corresponding to the endpoint of the temporal drawn image sequences, under the following conditions:

- RAW-IMG: Raw pixels are used as inputs in this analysis.
- DNN-IMG: Dimensionally compressed images featured by trained DNN are used for inputs.
- DNN–RNN: Initial values of Cs unit are used as inputs.

To acquire the PCA components in the case of RAW-IMG, the translated and rotated images are used to generalize the spatial variations. To translate and rotate the images, the same method as that for preparing DNN's training dataset is applied. The inputs only from the original training dataset are selected for the other two conditions.

Fig. 13 presents the results of PCA under the three conditions. The DNN-IMG features cluster better than RAW-IMG. On the other hand, DNN–RNN features are combined by each class more definitely than DNN-IMG. Although the acquired DNN-IMG feature structure is well organized compared with the RAW-IMG feature structure, circle and heart features are mixed in the space. For example, circle-3 and heart-3 are most likely paired. This similarity is probably due to the common visual characteristics shared by these shapes. Both these two shapes are mainly composed of a curved line and a straight line of the overdrawing parts. In contrast, in the case of DNN–RNN, these shapes are assigned quite different features because of the differences in drawing motions. In the DNN–RNN, circle-3 and heart-3 exhibit differences, but heart-1 and moon-3 are more similar than in DNN-IMG. This result is probably caused by the similarity of the learnt drawing motions by the MTRNN depicted in Fig. 14. The point of pause in the middle of heart-1's drawing motion is not reconstructed by the MTRNN, because the MTRNN represents the training drawing motions as perpetually changing sequences.

The classes' covariances, corresponding to each shape type, are compared to evaluate how well the training drawn images are organized by shape type. The covariances are obtained by the ratios of the between-class covariances and the within-class covariances

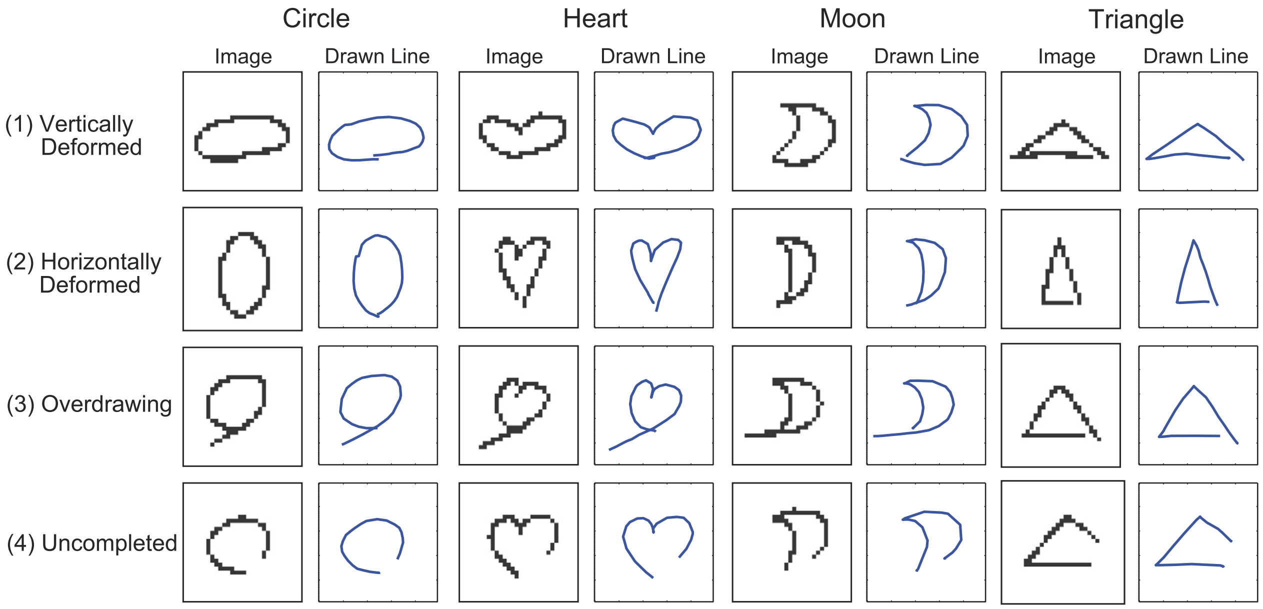


Fig. 11. The training dataset for the second experiment. There are four types of shapes and each type includes four distortion variations.

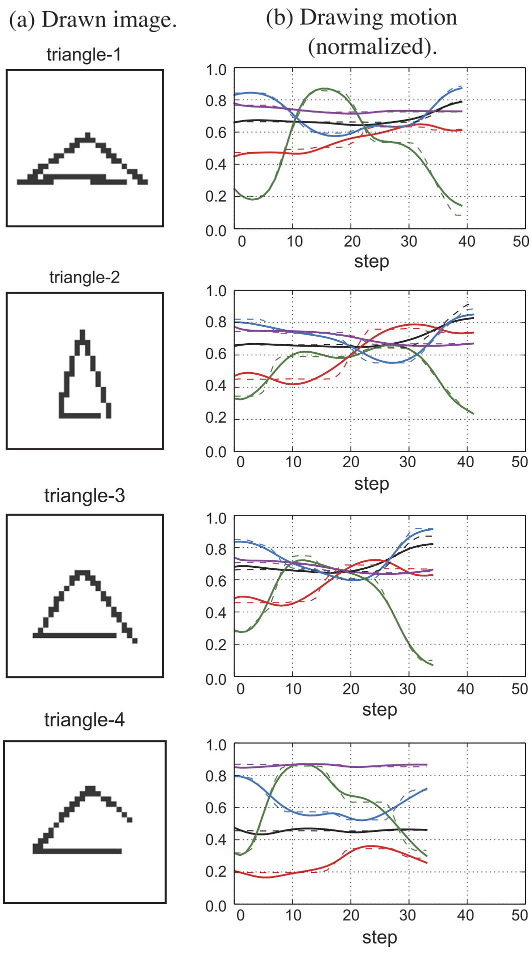


Fig. 12. Examples of the generated drawing sequences by the trained model; (a) Drawn images of the recorded training dataset; (b) Temporal feature of drawn images; (c) Time-series joint angles of the robot (solid lines are the generated angles by the MTRNN and dot lines are the training sequences). The last number in the shape's name corresponds to the type of distortions shown in Fig. 11.

Table 3
Class covariances. s_w is the within-class covariance, s_b is the between-class covariance, and S is the ratio of s_w and s_b.

	s_w	s_b	S
RAW-IMG	0.18	0.01	0.05
DNN-IMG	0.19	0.03	0.17
DNN-RNN	**0.19**	**0.11**	**0.56**

as follows:

$$s_w = \frac{1}{N}\sum_{i\in class}\sum_{m_i\in m}(m-\overline{m_i})^{\mathrm{T}}(m-\overline{m_i}) \qquad (14)$$

$$s_b = \frac{1}{N}\sum_{i\in class}(\overline{m_i}-\overline{m})^{\mathrm{T}}(\overline{m_i}-\overline{m}) \qquad (15)$$

$$S = \frac{s_b}{s_w}, \qquad (16)$$

where s_w is the within-class covariance, calculated using input feature m and averaging all the features in each class $\overline{m_i}$; s_b is the between-class covariance, using two means $\overline{m_i}$ and their mean $\overline{m}$; and S is the dimensionless number, which expresses how well the features converge or not by shape type. Table 3 summarizes the obtained covariances between the shape types. The largest ratio of covariances S refer to DNN–RNN, followed in order by DNN and RAW-IMG. In particular, the between-class covariances s_b contribute to these differences, which is the mean separation degree between the different types of shapes. Consequently, the DNN–RNN features are organized by the shape type better than those of DNN-IMG and RAW-IMG. These results clearly show that distorted drawn shapes are classified by the difference of not only visual characteristics of the drawn pictures, but also temporal drawn image features and drawing motions.

4. Discussion

4.1. Bottom-up approach for implementing a computational model of VM

The contribution of the present study is that the proposed model becomes the first case of bottom-up model which aims

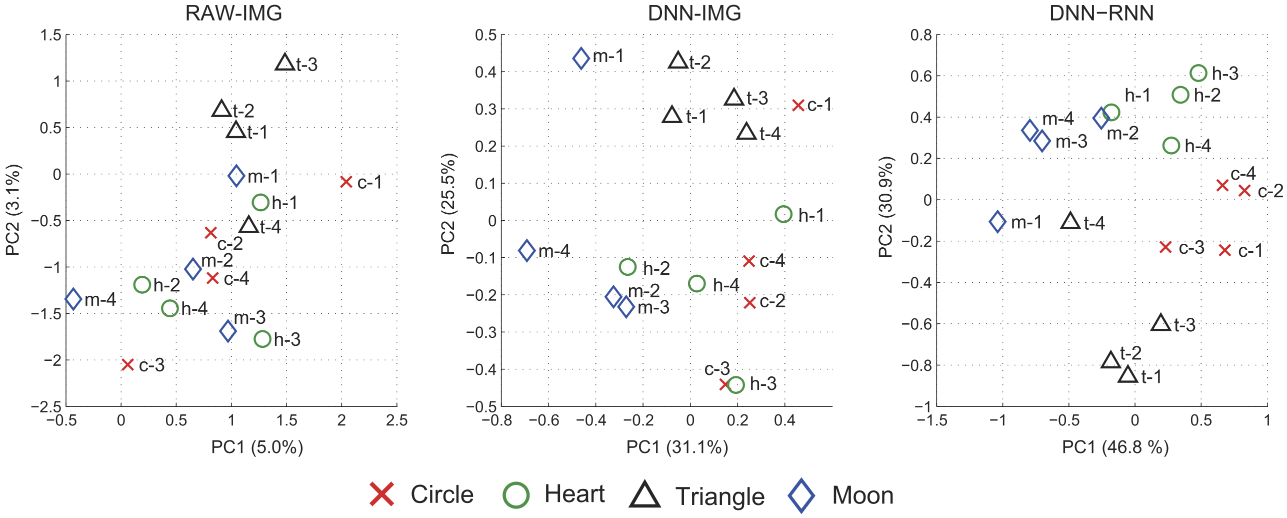


Fig. 13. The results of principal component analysis: In each plot, PC1 and PC2 axes correspond to principal components 1 and 2 with contribution values, respectively. The last number in the shape's name corresponds to the type of distortions shown in Fig. 11.

to replicate the human's drawing ability of using dynamical relationship between temporal drawn pictures and drawing motion. As mentioned in Section 1, the problem in implementing computational models that can learn both temporal drawn images and motions is the large calculation cost of processing images. In the present study, DNN is applied to solve this problem. This network was used to compress the sparse but large-dimensionality images into the low-dimensional features which can be integrated with the drawing motions. Further, MTRNN organizes the temporal drawing experiences into lower-dimensional features as the values of the Cs unit because of the hierarchical connectivity of the two types of recurrent layers. Consequently, these two strategies realize self-organized learning of robot's drawing sequences.

4.2. Association of drawing motion from drawn picture

One of the difficulties in investigating drawing ability is that drawn pictures contain many variations, even if certain meaning of pictures is shared with others. Because of this diversity of variations, it is impossible to make an assumption about prepositional knowledge of visual features which can cover whole variations of the picture. Cognitive researchers suggested that the solution lies in the observer's perception. These studies point out that the human ability to recognize shapes or letters relates to the observer's actual experiences of drawing or writing [1,2]. Consequently, it appears that the integrated feature of drawing sequences by the proposed model has a function similar to that of human's recognition system for pictures.

In the first experiment, we showed that the proposed model associates drawing motion with drawn images using organized memory from drawing experiences. In the second experiment, the model clearly disposes the acquired features of distorted shapes. The results of comparison with other conditions of the organizing method for drawing experiences demonstrate that the proposed model outperforms the conditions that require only visual information. These results suggest that motion data work effectively to distinguish shapes that do not share visual characteristics. Distinguishing hand-drawn pictures is challenging in computer vision because it is difficult to design appropriate features that can cover many variations in distortion. Learning many examples of hand-drawn images by supervised learning algorithms is suggested [29,30]. In these studies, they start to use the information of drawing order as the input data in order to improve recognition accuracy [31]. The present study also focuses on the temporal information in drawing process, but in another modality to form drawing pictures.

4.3. Limitation of the proposed model

The proposed model has the ability to adapt to different types of drawing sequences because it does not assume any shape information of the pictures. However, the model cannot learn a very large number of drawing sequences due to the network's limited capacity. This limitation drives from not the DNN, but the MTRNN. This is because temporal drawn images are very simple, whereas photorealistic images, which are a common target of DNN, are more complex. The capability of MTRNN could be enhanced by adding neurons in the context units because the space in MTRNN is expanded due to the increase of weights. However, the capability of the model might reach a plateau when we want to enhance the model toward more complex drawings, which include multiple lines. To deal with this problem, MTRNN could be generalized for complex drawing sequences by decomposing each sequence into primitive line drawings. Due to the hierarchical structure in the context units, MTRNN operates as a combination of primitive functions [32]. Another limitation of the proposed model is generating drawing motions, which include many discontinuous changing points, i.e., spiky edges. Due to the characteristics of the CTRNN, the model tends to generate trajectories by continuously changing dynamics. Therefore, the generated trajectories by the model have distortions in the edge points. This can be resolved by adding pauses in direct teaching [17], or by utilizing the other type of recurrent neural networks, which is superior in learning discontinuous sequences, e.g., Long Short-Term Memory [33].

5. Conclusion

In this paper, we proposed a neural network based learning model for integration of drawn picture and motion in the robot's drawing experiences. The model is designed for acquiring VM, which replicates human's VM in the two aspects: (1) VM is formed by bottom-up learning for integration of temporal drawn picture and motion and (2) VM allows the observers to associate drawing motion from a picture.

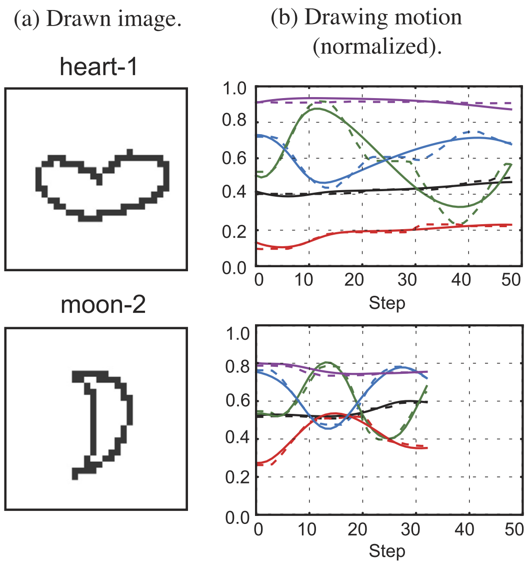


Fig. 14. Examples of the generated drawing sequences by the trained model; (a) Drawn images of the recorded training dataset; (b) Time-series joint angles of the robot (solid lines are the generated angles by the MTRNN, and dot lines are the training sequences). The last number in the shape's name corresponds to the type of distortions shown in Fig. 11.

To integrate temporal drawn images and motion, the model utilizes DNN for acquiring dimensionally compressed feature of the image. After compressing images, MTRNN is trained to conduct integrative learning of the acquired image feature and motion. After the learning process, the proposed model associates drawing motion which will represent the target picture by the generated drawing motion though the regression of the initial state of MTRNN.

Assuming that the robot draws simple pictures with a single stroke, two experiments are conducted to demonstrate learnt VM function by the model. In the first experiment, the proposed model successfully acquired the organized memory of 15 drawing sequences, including three types of shapes without prior knowledge of the shapes' visual features. Further, the ability to associate drawing motions from a drawn image is tested by three untrained shapes. Although the association drawing results include distortions, the generated drawing motions presented the similarity with motion of the same type of shape. In the second experiment, we also presented the motor-perceptual ability of the proposed model for four image types whose shapes are distorted into 16 different shapes. As the result, the acquired features by the proposed model were clearly divided according to shape with utilizing drawn motions.

Future work is required to investigate pictures drawn with multiple strokes, thus including a wider range of pictures. As mentioned in Section 1, the temporal order of strokes affects the recognition of letters [3]. Therefore, it is possible to apply the VM in more general drawing situations. A neuropsychological study suggests that motor-planning functions are utilized for pictures drawn with multiple strokes [34]. In addition, to expand the image learning part into more complex and high-resolution images, specified models for efficient image learning may be applied, such as convolutional neural networks [35].

Acknowledgments

The work has been supported by MEXT Grant-in-Aid for Scientific Research (A) 15H01710, and Scientific Research on Innovative Areas "Constructive Developmental Science" 24119003.

References

[1] J. Freyd, Representing the dynamics of a static form, Mem. Cogn. 11 (4) (1983) 342–346.

[2] M. Babcock, J. Freyd, Perception of dynamic information in static handwritten forms, Amer. J. Psychol. 101 (1) (1988) 111–130.

[3] J. Parkinson, B. Khurana, Temporal order of strokes primes letter recognition, Q. J. Exp. Psychol. 60 (9) (2007) 1265–1274.

[4] A. Pignocchi, How the intentions of the draftsman shape perception of a drawing, Consci. Cogn. 19 (4) (2010) 887–898.

[5] A.H. Waterman, J. Havelka, P. Culmer, L. Hill, M. Mon-Williams, The ontogeny of visual motor memory and its importance in handwriting and reading: a developing construct, Proc. R. Soc. B 282 (1798).

[6] B. Hommel, J. Musseler, G. Aschersleben, W. Prinz, The theory of event coding (tec): a framework for perception and action planning, Behav. Brain Sci. 24 (2001) 849–937.

[7] S. Calinon, J. Epiney, A. Billard, A humanoid robot drawing human portraits, in: Proceedings of the 5th IEEE-RAS International Conference on Humanoid Robots, Tsukuba, Japan, 2005, pp. 161–166.

[8] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (1986) 679–698.

[9] S. Kudoh, K. Ogawara, M. Ruchanurucks, K. Ikeuchi, Painting robot with multi-fingered hands and stereo vision, Robot. Auton. Syst. 57 (3) (2009) 279–288.

[10] S. Mueller, N. Huebel, W. Waibel, R. D'Andrea, Robotic calligraphy –learning how to write single strokes of chinese and japanese characters, in: Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 2013, pp. 1734–1739.

[11] H.M.L. Josh, Y. Yam, Stroke trajectory generation experiment for a robotic chinese calligrapher using a geometric brush footprint model, in: Proceedings of 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, USA, 2009, pp. 2315–2320.

[12] O. Deussen, T. Lindemeier, S. Pirk, M. Tautzenberger, Feedback-guided stroke placement for a painting machine, in: Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging, 2012, pp. 25–33. http://dx.doi.org/10.2312/COMPAESTH/COMPAESTH12/025-033.

[13] P. Tresset, F.F. Leymarie, Portrait drawing by Paul the robot, Comput. Graph. 37 (5) (2013) 348–363.

[14] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, C. Yoshida, Cognitive developmental robotics: A survey, IEEE Trans. Auton. Mental Dev. 1 (1) (2009) 12–34.

[15] V. Mohan, P. Morasso, J. Zenzeri, G. Metta, V. Chakravarthy, G. Sandini, Teaching a humanoid robot to draw 'shapes', Auton. Robots 31 (1) (2011) 21–53.

[16] K. Mochizuki, S. Nishide, H. Okuno, T. Ogata, Developmental human–robot imitation learning of drawing with a neuro dynamical system, in: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, Manchester, England, 2013, pp. 2336–2341.

[17] S. Nishide, K. Mochizuki, H. Okuno, T. Ogata, Insertion of pause in drawing from babbling for robot's developmental imitation learning, in: Proceedings of IEEE International Conference on Robotics and Automation, Hong Kong, China, 2014, pp. 4785–4791.

[18] A. Droniou, S. Ivaldi, O. Sigaud, Deep unsupervised network for multimodal perception, representation and classification, Robot. Auton. Syst. 71 (2015) 83–98.

[19] K. Noda, H. Arie, Y. Suga, T. Ogata, Multimodal integration learning of robot behavior using deep neural networks, Robot. Auton. Syst. 62 (6) (2014) 721–736.

[20] R. Beer, On the dynamics of small continuous-time recurrent neural networks, Adapt. Behav. 3 (4) (1995) 469–510.

[21] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[22] P. Werbos, Backpropagation through time: What it does and how to do it, Proc. IEEE 78 (10) (1990) 15550–15560.

[23] D.E. Rumelhart, J.L. McClelland, P.R. Group, Parallel Distributed Processing Volume 1, Explorations in the Microstructure of Cognition, MIT Press, 1986.

[24] J. Martens, Deep learning via hessian-free optimization, in: Proceedings of 27th International Conference on Machine Learning, vol. 951, Haifa, Israel, 2010, pp. 735–742.

[25] N. Schraudolph, Fast curvature matrix–vector products for second-order gradient descent, Neural Comput. 14 (7) (2002) 1723–1738.

[26] Y. Yamashita, J. Tani, Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment, PLoS Comput. Biol. 4 (11).

[27] A. Robotics, Nao humanoid [online] (July 2015). http://doc.aldebaran.com/2-1/home_nao.html.

[28] Wacom, Intuous pen & touch small [online] (July 2015). http://www.wacom.com/en-us/products/pen-tablets/intuos-pen-and-touch-small.

[29] M. Eitz, J. Hays, M. Alexa, How do humans sketch objects? ACM Trans. Graph. 31 (4) (2012) 1–10. http://dx.doi.org/10.1145/2185520.2335395.
URL http://dl.acm.org/citation.cfm?doid=2185520.2335395.

[30] Q. Yu, Y. Yang, F. Liu, Y. Song, Xiang, T. Hospedales, Int. J. Comput. Vis. (2016) 1–15. http://dx.doi.org/10.1007/s11263-016-0932-3.

[31] Q. Yu, F. Liu, Y. Yang, T. Xiang, T.M. Hospedales, C.C. Loy, Sketch me that shoe, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. http://dx.doi.org/10.1007/s11263-016-0932-3.

[32] A. Hiroaki, A. Takafumi, S. Shigeki, T. Jun, Imitating others by composition of primitive actions: A neuro-dynamic model, Robot. Auton. Syst. 60 (5) (2012) 729–741. http://dx.doi.org/10.1016/j.robot.2011.11.005.

[33] A. Graves, Supervised Sequence Labelling with Recurrent Neural Networks, Studies in Computational Intelligence and Complexity, Springer, 2012.

[34] S. McCrea, A neuropsychological model of free-drawing from memory in constructional apraxia: A theoretical review, Amer. J. Psychiatry Neurosci. 2 (5) (2014) 60–75.

[35] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in 25th Conference on Neural Information Processing Systems, Harrah's Lake Tahoe, US, 2012, pp. 1097–1105.

Kazuma Sasaki received the B.S. and M.S. degrees in Engineering in 2013 and 2015, respectively, from Waseda University, Japan. Currently, he is a Ph.D. candidate at Waseda University and in the graduate program for Embodiment Informatics, a part of the programs for the leading graduate schools of Ministry of Education, Culture, Sports, Science and Technology Japan. His research interests include human's drawing ability, sketch recognition system, deep learning, and autonomous robots.

Kuniaki Noda received the B.S. and M.S. degrees in Mechanical Engineering in 2000 and 2002, respectively, from Waseda University, Japan. From 2002 to 2013, he worked for Sony Corporation. From 2009 to 2010, he was a visiting researcher at EPFL, Switzerland. Currently, he is a Ph.D. candidate at Waseda University. His research interests include autonomous robot, multimodal integration, deep learning, and high performance computing on GPU. He received various awards including the Hatakeyama Award from the Japan Society of Mechanical Engineers in 1999, the Best Paper Award of ICDL-EPIROB 2011, and the Best Paper Award of RSJ in 2012.

Tetsuya Ogata received the B.S., M.S. and D.E. degrees in Mechanical Engineering, in 1993, 1995 and 2000, respectively, from Waseda University. He was a Research Fellow of JSPS, a Research Associate of Waseda University, a Research Scientist of RIKEN Brain Science Institute, and an Associate Professor of Kyoto University. He is currently a Professor of Faculty of Science and Engineering, Waseda University. His research interests include neural models for robots, dynamics of human–robot mutual adaptation and inter-sensory translation. He is a member of IEEE, RSJ, JSAI, IPSJ, JSME, SICE, etc.