///VSP///

Advanced Robotics 25 (2011) 825–848

ADVANCED ROBOTICS
brill.nl/ar

Full paper

Learning, Generation and Recognition of Motions by Reference-Point-Dependent Probabilistic Models

Komei Sugiura *, **Naoto Iwahashi**, **Hideki Kashioka and Satoshi Nakamura**

National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika, Soraku, Kyoto 619-0289, Japan

Received 15 December 2009; accepted 1 April 2010

Abstract

This paper presents a novel method for learning object manipulation such as rotating an object or placing one object on another. In this method, motions are learned using reference-point-dependent probabilistic models, which can be used for the generation and recognition of motions. The method estimates (i) the reference point, (ii) the intrinsic coordinate system type, which is the type of coordinate system intrinsic to a motion, and (iii) the probabilistic model parameters of the motion that is considered in the intrinsic coordinate system. Motion trajectories are modeled by a hidden Markov model (HMM), and an HMM-based method using static and dynamic features is used for trajectory generation. The method was evaluated in physical experiments in terms of motion generation and recognition. In the experiments, users demonstrated the manipulation of puppets and toys so that the motions could be learned. A recognition accuracy of 90% was obtained for a test set of motions performed by three subjects. Furthermore, the results showed that appropriate motions were generated even if the object placement was changed.

Keywords

Imitation learning, robot language acquisition, hidden Markov model, object manipulation

1. Introduction

Adaptability and scalability have been key issues in the realm of machine learning. In robotics, many studies on the applications of machine learning have been conducted to develop machines able to adapt to various environments and to acquire new skills. Imitation learning, which explores methods to teach a robot to perform new tasks by showing them, is one approach adopted in such studies [1–3]. Robotics research on imitation learning started in the 1990s under names such as learning by watching, learning from demonstration and programming by demonstration [3, 4].

* To whom correspondence should be addressed. E-mail: komei.sugiura@nict.go.jp

From the perspective of scalability, programming humanoid robots to perform new tasks is an attractive application of imitation learning since the cost of designing a large number of control parameters can be reduced [1].

From the perspective of adaptability, imitation learning has another importance. Object-manipulating motions such as ‘put the dishes in the cupboard’ are fundamental for robots aimed at home environments, but difficult to program beforehand. This is because the desired motion depends on elements specific to each home: the size and shape of the dishes and the cupboard, and whether the cupboard has a door. In contrast, if a robot can learn motions from multiple observations of the user’s motions, obtaining a controller for motions adapted to each home is possible. Moreover, it is important to note that the meaning of learned motions is shared by the user and the robot in this case. Thus, imitation learning can be a mechanism for situated communication since the shared meaning is grounded on the sensory-motor experiences of them [1]. Situated communication between a human and a robot in object-manipulating tasks has been attempted in Refs [5, 6].

One of the main difficulties in learning object manipulation is that clustering trajectories in a fixed coordinate system is not effective, since the trajectories are then not reusable in other conditions. An example of this is learning to ‘place an object on another’. Initial object placements are usually not fixed within the user’s demonstrations and, in addition, the robot has to generate appropriate motions under various object placements. Therefore, it is necessary to generalize relative motions between objects in the learning phase and synthesize an appropriate motion trajectory for the given object placement in the generation phase.

Some recent studies have attempted to solve the difficulties in learning object manipulation [4]. Regier investigated a model describing the spatial relationship between two objects [7]. He proposed to model motions as the time evolution of the spatial relationship between a trajector and a landmark. Ogawara *et al.* attempted to model the relative trajectories between two objects by using hidden Markov models (HMMs) [8, 9]. In their studies, input from a data glove and relative trajectories between a cup and a bowl were used, and learned motions such as ‘grasping’ and ‘pouring’ were performed by a robot. Billard *et al.* developed a method that extracted important features from task and joint spaces by introducing a cost function, which was the weighted sum of task- and joint-space-based reproduction errors. In Ref. [10], relative trajectories were modeled by HMMs and optimal state numbers were determined by model selection. Calinon *et al.* used multiple Gaussian distributions to model the time series of spatiotemporal values [11]. Manipulation trajectories by a couple of robot arms were generated based on Gaussian Mixture Regression (GMR).

These methods use relative trajectories between two objects under an assumption that the two objects are specified by the user. Therefore, there is a limitation that the user has to specify which motion needs information on the two objects and which does not. For example, the positions of two objects are necessary for learning the ‘place an object on another’ motion, while relative trajectories do not have to be considered in learning the 'raise an object' motion. Thus, overcoming this limitation enables the learning of various types of object manipulation from natural interactions between the user and robot.

The objective of this study is to construct a method for learning object manipulation such as placing an object on another (place-on) or moving it away (move-away). The proposed method has three key features:

(i) The reference point and the type of intrinsic coordinate system, which is the type of coordinate system intrinsic to a motion, are estimated based on a maximum likelihood criterion. In this process, the number of objects necessary to describe the motion is determined and the relevant object(s) in each demonstration is estimated as well.

(ii) Manipulation trajectories are modeled by HMMs using dynamic features as well as static features.

(iii) The optimal number of states of the HMMs is estimated by cross-validation.

With the above first feature, the proposed method can avoid suffering from the limitation that the user has to specify the relative objects for the motion. Therefore, it can deal with motions such as raising an object and placing an object on another using the same framework. The second feature has several advantages over other methods (e.g., Ref. [12]) such as (i) learning object-manipulation motion that is dependent on velocity is possible, and (ii) the maximum likelihood trajectory can be searched effectively [13]. Although the maximum likelihood trajectory can be generated from a position-only HMM (HMMs with static features), some sort of interpolation or smoothing is needed since the trajectory is discontinuous. Such trajectory is not guaranteed to be the maximum likelihood trajectory any more.

Figure 1 shows the hardware platform used in this study. The system has multimodal interfaces such as a stereo vision camera and a microphone. Online and offline learning are possible with the system. In the online learning mode, a user demonstrates motion trajectories with uttering the corresponding motion label. In this paper, however, we show experimental results obtained with the offline mode

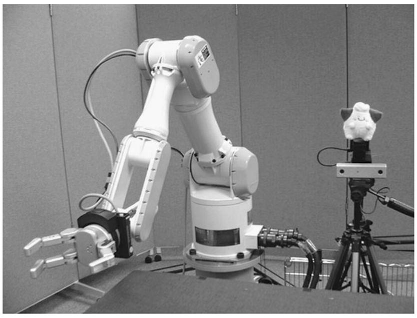

Figure 1. Hardware platform used in the experiments.

since the focus of the paper is on the quantitative evaluation of the proposed method by dividing samples into training and test sets.

The rest of this paper is organized as follows. Section 2 first states the basic concepts and terminology for reference-point-dependent motions. Section 3 describes the proposed method. The experimental setup and results for the learning, generation and recognition of motions are presented in Sections 4, 5 and 6, respectively. Section 7 discusses the extensions and problems of the proposed method. Section 8 states related work and Section 9 concludes the paper.

2. Learning Reference-Point-Dependent Motions

In cognitive linguistics, a trajector is defined as a participant (object) being focused on. A landmark has a secondary focus and a trajector is characterized with respect to the landmark. Words representing spatial relationships such as 'away' and 'left of' are described in terms of a relationship between a trajector and a landmark [14]. An example of this is shown in the left-hand panel of Fig. 2. This depicts a camera image in which the green puppet (Kermit) is moved along the dotted line. In the example shown in Fig. 2, the trajector is Kermit. On the other hand, there are multiple interpretations of the landmark. If the blue (right) box is considered as the landmark, the label 'place-on' can be provided to the trajectory. However, the label can become 'Let Kermit jump over the green box (jump-over)' if the green (left) box is thought of as the landmark.

Motions such as 'place-on' and 'jump-over' are dependent on landmarks. Other motions, such as 'raise an object', do not depend on a landmark; however, this motion means that the object is moved higher than its original position. Therefore, 'place-on' and 'raise' are both modeled as motions dependent on particular references. In this paper, those motions are called reference-point-dependent motions.

Here, the problem of learning reference-point-dependent motions in a learning-from-demonstration framework is considered. The reference point of each demonstration is not specified by the user in the learning phase. This setup is based on the fact that it is not always possible to specify the reference point for the given object placement only through linguistic expressions. Moreover, an easy-to-use interaction

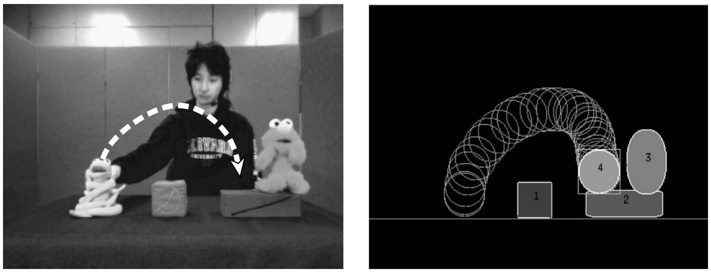

Figure 2. (Left) Example shot of an image stream. (Right) Preprocessed visual features obtained from the image stream.

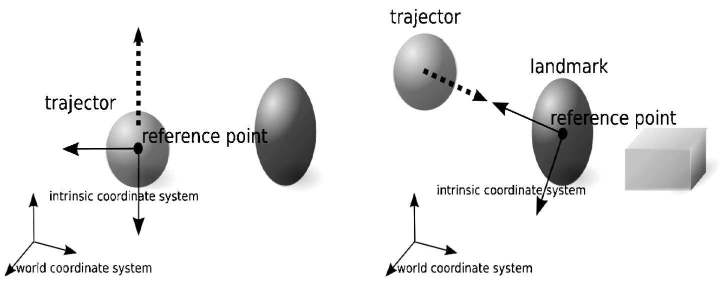

Figure 3. Relationship between a trajector/landmark, a reference point and an intrinsic coordinate system. The spheres, ellipsoids and box represent objects, and the arrows represent the axes of the intrinsic coordinate systems. (Left) ‘Raise’. (Right) ‘Move-closer’.

of teaching motions is possible by using this setup. Another unspecified element is the intrinsic coordinate system type. Object manipulation involves a spatial relationship between objects, and it is important to select an appropriate intrinsic coordinate system, which is the frame of reference [15] intrinsic to a motion, to describe the evolution of the spatial relationship.

Two examples of this can be shown by ‘raise’ and ‘move-closer’ (Fig. 3). The reference point of ‘raise’ can reasonably be assumed to be the trajector’s center of gravity. The intrinsic coordinate system used can be a Cartesian coordinate system, as shown in the left-hand panel. However, in the case of ‘move-closer’, another type of intrinsic coordinate system is necessary. In this case, the x-axis of the coordinate system passes through the centers of gravity of the trajector and the landmark. The point is that the reference point and intrinsic coordinate system are unobservable in the problem trying to be solved, and so they must be estimated.

3. Learning from Observation by Reference-Point-Dependent Probabilistic Models

3.1. Estimation of Reference Points and Intrinsic Coordinate Systems

Consider that L training samples are given for a kind motion such as ‘place-on’ or ‘raise’. Let $\mathcal{V}_l$ denote the lth training sample. $\mathcal{V}_l$ consists of the motion information of the trajector, $\mathcal{Y}_l$, and the set of positions of the static objects, $\mathbf{O}_l$, as:

$$\mathcal{V}_l = (\mathcal{Y}_l, \mathbf{O}_l) \tag{1}$$

$$\mathcal{Y}_l = \{\mathbf{y}_l(t) | t = 0, 1, \ldots, T_l\} \tag{2}$$

$$\mathbf{y}_l(t) = \left[\mathbf{x}_l(t)^\top, \dot{\mathbf{x}}_l(t)^\top, \ddot{\mathbf{x}}_l(t)^\top\right]^\top, \tag{3}$$

where $\mathcal{Y}_l$ is a time series of vectors $\mathbf{y}_l(t)$. Here, $\mathbf{y}_l(t)$ is composed of position $\mathbf{x}_l(t)$, velocity $\dot{\mathbf{x}}_l(t)$ and acceleration $\ddot{\mathbf{x}}_l(t)$. For simplicity, the index l is omitted when the context is clear.

It is assumed that there are several types of intrinsic coordinate systems and these are given by the designer. The type of the intrinsic coordinate system is denoted by k. A schematic illustration is shown in Fig. 3 and such types are explained in detail in Section 4.2.

From the estimation of k and the reference point $\mathbf{x}^r$, the intrinsic coordinate system is obtained in the lth data. The candidate set of reference points is denoted by $\mathbf{R}$. The set of the positions of static objects, $\mathbf{O}$, has to be included in $\mathbf{R}$, since these objects are the candidates for the landmark. The first position of the trajector, $\mathbf{x}(0)$, is included in $\mathbf{R}$ so that a motion concept that is dependent only on the object's trajectory can be described. In addition, the center of the camera image, $\mathbf{x}_{\text{center}}$, is put in $\mathbf{R}$ to describe motion concepts independent from the positions of objects. Therefore:

$$\mathbf{R} = \{\mathbf{O}, \mathbf{x}(0), \mathbf{x}_{\text{center}}\} \triangleq \{\mathbf{x}^r \mid r = 1, 2, \ldots, |\mathbf{R}|\}, \tag{4}$$

where $|\mathbf{R}|$ denotes the number of elements contained in $\mathbf{R}$.

Let ${}^{C_k(\mathbf{x}^{r_l})}\mathcal{Y}_l$ denote the trajectory in the intrinsic coordinate system $C_k(\mathbf{x}^{r_l})$, where r_l is the index of the reference point in $C_k(\mathbf{x}^{r_l})$. Henceforth, a parameter in a specific coordinate system is represented by using such left superscript. Now, the optimal k, the optimal set of reference points, $\mathbf{r}$, and the optimal parameters of a probabilistic model, λ, are searched for using the following maximum likelihood criterion:

$$(\hat{\lambda}, \hat{k}, \hat{\mathbf{r}}) = \underset{\lambda,k,\mathbf{r}}{\arg\max} \sum_{l=1}^{L} \log P(\mathcal{Y}_l | r_l, k, \lambda) \tag{5}$$

$$= \underset{\lambda,k,\mathbf{r}}{\arg\max} \sum_{l=1}^{L} \log P\left({}^{C_k(\mathbf{x}^{r_l})}\mathcal{Y}_l; \lambda\right), \tag{6}$$

where $\hat{\cdot}$ represents estimation.

The number of candidate solutions for k is generally smaller than for $\mathbf{r}$, which is $\prod_{l=1}^{L} |\mathbf{R}_l|$, so we apply a brute-force method for obtaining the optimal k. Below, the optimal λ and $\mathbf{r}$ are searched for each k.

Thus, we obtain the following equation from (6):

$$(\hat{\lambda}^k, \hat{\mathbf{r}}^k) = \underset{\lambda,\mathbf{r}}{\arg\max} \sum_{l=1}^{L} \log P\left({}^{C_k(\mathbf{x}^{r_l})}\mathcal{Y}_l; \lambda\right), \tag{7}$$

where $\cdot^k$ means that k is fixed. It is not practical to solve (7) due to its computational complexity. Therefore, for constraint relaxation, the above discrete optimization problem is approximated as a continuous optimization problem:

$$(\hat{\lambda}^k, \hat{\mathbf{w}}^k) = \underset{\lambda,\mathbf{w}}{\arg\max} \sum_{l=1}^{L} \log \left[\sum_{r_l=1}^{|\mathbf{R}_l|} w_{l,r_l}^k P\left({}^{C_k(\mathbf{x}^{r_l})}\mathcal{Y}_l; \lambda\right) \right], \tag{8}$$

where:

$$\sum_{r_l=1}^{|\mathbf{R}_l|} w^k_{l,r_l} = 1 \quad (l = 1, 2, \ldots, L)$$
$$\mathbf{w}^k_l = \left(w^k_{l,1}, w^k_{l,2}, \ldots, w^k_{l,|\mathbf{R}_l|}\right)$$
$$\mathbf{w}^k = \left(\mathbf{w}^k_1, \mathbf{w}^k_2, \ldots, \mathbf{w}^k_L\right).$$

Here, w^k_{l,r_l} denotes the weight for selecting the reference point r_l in the training sample $\mathcal{V}_l$ where k is fixed. Thus, $\mathbf{r}$ can be thought of as a set of hidden parameters. Equation (8) can efficiently be solved by applying an EM algorithm [16]. In the Appendix, a solution to (8) is explained when an HMM is used as the probabilistic model. The optimal k is obtained by solving:

$$\hat{k} = \arg\max_k \sum_{l=1}^{L} \log\left[\sum_{r_l=1}^{|\mathbf{R}_l|} \hat{w}^k_{l,r_l} P\left({}^{C_k(\mathbf{x}^{r_l})}\mathcal{Y}_l; \hat{\lambda}^k\right)\right]. \tag{9}$$

To estimate the optimal number of states of HMMs, leave-one-out cross-validation [17] was used. In the first stage of the cross-validation process, training samples were divided into the real training set ($L - 1$ samples) and the validation set (one sample).

A HMM was then trained by using the real training set and evaluated with the validation set. The above procedure was repeated for possible combinations of training sets and validation sets, and the optimal number that gave the maximum average likelihood was selected.

3.2. Motion Generation by Reference-Point-Dependent HMMs

Here we consider the problem of generating trajectories from reference-point-dependent probabilistic models obtained according to the aforementioned method. Below, an HMM is used as a reference-point-dependent probabilistic model.

Let $\mathbf{q} = \{q_t \mid t = 0, 1, \ldots, T\}$ denote a sequence of states, which are unobservable. Now, consider the problem of obtaining a trajectory that maximizes the probability of $\mathcal{Y}$ conditioned by the position of a trajector $\mathbf{x}^{r_{\text{traj}}}$, $\mathbf{q}$, motion label, v_i, and reference point index, r, as:

$$\hat{\mathcal{Y}} = \arg\max_{\mathcal{Y}} P\left(\mathcal{Y} | \mathbf{x}^{r_{\text{traj}}}, \mathbf{q}, r, v_i, \mathbf{R}\right) \tag{10}$$
$$= \arg\max_{\mathcal{Y}} P\left(\mathcal{Y} | \mathbf{x}^{r_{\text{traj}}}, \mathbf{q}, r, k_i, \lambda_i, \mathbf{R}\right). \tag{11}$$

$\mathbf{x}^{r_{\text{traj}}}$ is put in the condition since $\hat{\mathcal{Y}}$ has to start from $\mathbf{x}^{r_{\text{traj}}}$. $\mathbf{x}^{r_{\text{traj}}}$ is obtained from the index of the trajector and $\mathbf{R}$. The parameters of the HMM corresponding to the ith verb are denoted by $\lambda_i = (\pi, A, \mathbf{b})$, where π, A and $\mathbf{b}$ denote the initial probability distribution, the state transition probabilities and the output probability density functions, respectively.

When a left-to-right HMM is used and T is given, (11) can be solved by applying Tokuda's method [13]. The method gives the maximum likelihood trajectory based on both static and dynamic features. As a matter of practical convenience, the trajectory is generated in the intrinsic coordinate system and then transformed into the camera/world coordinate system.

3.3. Motion Recognition by Reference-Point-Dependent HMMs

Next we consider motion recognition by reference-point-dependent HMMs. Let $\mathcal{V}$ denote the observed information. Similar to the above, $\mathcal{V}$ consists of the trajectory of the moving object, $\mathcal{Y}$, and the set of positions of static objects, $\mathbf{O}$. Motion recognition is formulated as a problem of obtaining the maximum likelihood probabilistic model to output $\mathcal{Y}$.

Let $V = \{v_i \mid i = 1, 2, \ldots, |V|\}$ denote a set of learned motion labels, λ_i denote the HMM parameter set that corresponds to motion label v_i, and k_i denote the index of the intrinsic coordinate system of v_i. (λ_i, k_i) is obtained by using the aforementioned learning method. The maximum likelihood pair of the motion label and reference point indices, $(\hat{i}, \hat{r})$, are searched for as:

$$(\hat{i}, \hat{r}) = \arg\max_{i,r} P(\mathcal{Y}|r, v_i, \mathbf{R}) \tag{12}$$

$$= \arg\max_{i,r} P(\mathcal{Y}|r, k_i, \lambda_i, \mathbf{R}). \tag{13}$$

4. Experiment 1: Motion Learning

4.1. Hardware Platform

The experiments were conducted with the platform shown in Fig. 1. The user's movements were recorded by a Bumblebee 2 stereo vision camera at a rate of 30 frame/s. The size of each camera image was 320×240 pixels. The left-hand panel of Fig. 2 shows an example shot of an image stream, and the right-hand panel shows the internal representation of the image stream. The transformation matrix between the camera coordinate system and the world coordinate system was fixed since the camera was fixed in the environment. The recorded data were used for the learning and recognition of motions.

Motion generation experiments were conducted with a Mitsubishi Heavy Industries PA-10 manipulator with 7 d.o.f. The manipulator was equipped with a Barrett Hand, a 4-d.o.f. multifingered grasper. Motion generation results were examined in an environment using the manipulator and physical objects such as puppets and toys. The hardware platform was used for motion recognition/generation experiments as well.

4.2. Experimental Setup

In this subsection, the setup of the experiments conducted to obtain trained reference-point-dependent HMMs for subsequent motion generation and recognition experiments is described.

A subject was asked to demonstrate motions that manipulated objects placed on a desk. Each motion was given to the subject in the form of a motion such as 'raise' or 'rotate'. The environment used in the experiments is illustrated in the left-hand panel of Fig. 2. A training sample was obtained from the manipulation by preprocessing the recorded image stream of the manipulation (see the right-hand panel of Fig. 2).

For the tracking of objects, we used a heuristic algorithm, which tracks objects on/above the table between the user and robot. The start and end points of object trajectories were determined *a priori*. Specifically, if an object remained in the same position for five frames, the trajectory of the object was segmented. The object placement was changed after each demonstration.

All trajectories were obtained from the demonstration by the same subject, Subject A. The following motion labels (verbs) were used in the experiments: jump-over, move-away, move-closer, place-on, put-down, raise, rotate. The trajectories corresponding to each motion label were used to train the corresponding HMM. Seven different HMMs were obtained from the trajectory sets. The duration of each state of an HMM was also recorded. The average duration of each state was used when a trajectory was generated from an HMM.

The training sets obtained in the experiments are summarized in Table 1. TA-2 to TA-9 were obtained by increasing the number of training samples, L. The average number of objects contained in a sample in TA-9 was 3.1, which means a typical scene contained one trajector and two landmark candidates on average.

The following types of intrinsic coordinate systems were defined:

C_1 A translated camera coordinate system with its origin at the landmark position. The direction of the y-axis is always vertical. The x-axis is inverted in the case that the x coordinate of $\mathbf{x}_l(0)$ is negative after translation.

Table 1.
Training sets

Training set	Subject	No. of motion labels	No. of trajectories per motion label
TA-2	A	7	2
TA-3	A	7	3
⋮	⋮	⋮	⋮
TA-9	A	7	9

C_2 An orthogonal coordinate system with its origin at the landmark position. The direction of the x-axis is from the landmark position towards $\mathbf{x}_l(0)$.

C_3 A translated camera coordinate system with its origin at $\mathbf{x}_l(0)$.

C_4 A translated camera coordinate system with its origin at $\mathbf{x}_{\text{center}}$.

In C_3- or C_4-type intrinsic coordinate systems, the reference point does not have to be estimated since it is fixed to $\mathbf{x}_l(0)$ or $\mathbf{x}_{\text{center}}$. Therefore, the proposed estimation method (8) was not applied.

The weight for selecting the reference point r, w_l, was initialized to $w_l = 1/|O|$, where $|O|$ denotes the number of static objects. Note that l, k are omitted here for simplicity. The number of states for the initial HMMs was either of {6, 10, 14, 18}.

The average computation time for learning each motion was 9.8 s, where the training set of a motion contained nine samples. The computation was carried out on a quad-core 2.66-GHz computer running the Linux operating system. The HMM Toolkit (HTK [18]) was used for training HMMs. In the HTK, HMM parameters are initialized based on iterative application of the Viterbi algorithm, which follows the association of uniformly segmented training data with each state.

4.3. Results

Figure 4 illustrates examples of the internal representations for the training samples. In Fig. 4, motion labels are shown below the corresponding panels. The final positions of the trajectors are illustrated as the painted ellipses and the sequences of the contours represent the trajectories.

Figure 5 shows the output probability density functions (OPDFs) for HMMs trained with trajectories corresponding to the motion label ‘place-on’. The HMM for which the optimal state number was estimated as 18 was obtained using TA-9. The left-hand, middle and right-hand panels of Fig. 5 illustrate distributions corresponding to the position, velocity and acceleration, respectively. The center and semimajor (or semiminor) axis of each ellipse stands for the mean and standard deviation of each distribution. The direction of the state transition is indicated by the darkness of shading. From Fig. 5, the HMM is shown to have a large variance in the first state and small variance in the last state for position. This fact indicated that the trajectory of ‘place-on’ started from an arbitrary point, but converged to a certain point in the end.

The results regarding the estimation of the optimal state number by cross-validation are shown in Fig. 6. Trajectories regarding ‘place-on’ were transformed into coordinate system C_1 and used for training HMMs. Figure 6 plots the likelihood of four different HMMs against the number of training samples. In this paper, likelihood values shown in the figures are normalized by the number of frames. Instead of likelihood, $\mathcal{L}$, $-\log|\log \mathcal{L}|$ is shown so that the reader can easily find which HMM has the maximum likelihood. In Fig. 6, the average log-likelihood of different numbers of states are compared by cross-validation. Each line shows the

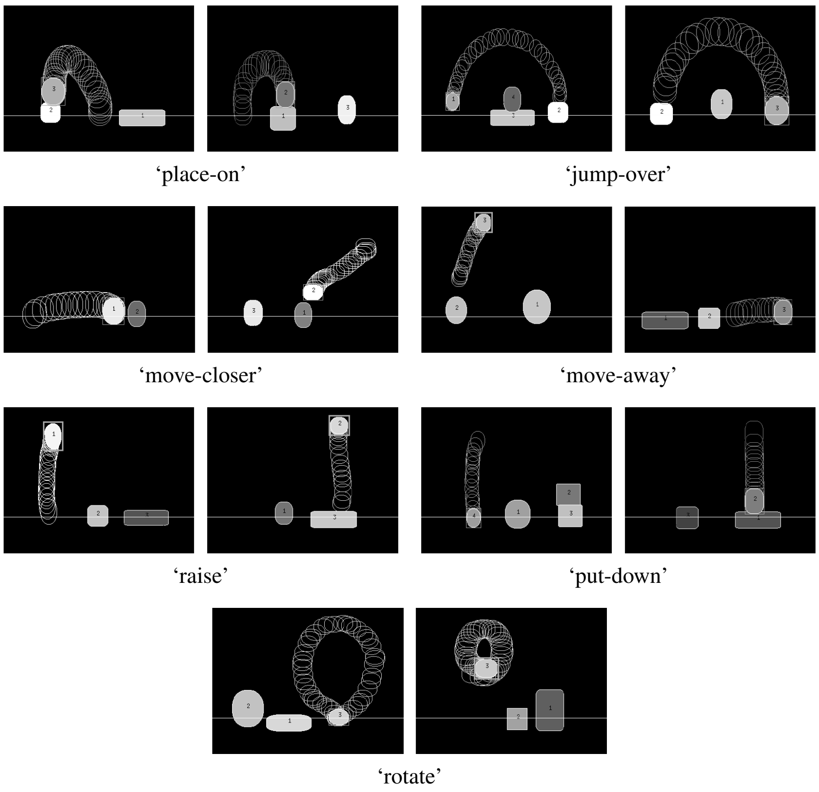

Figure 4. Example internal representation of training samples.

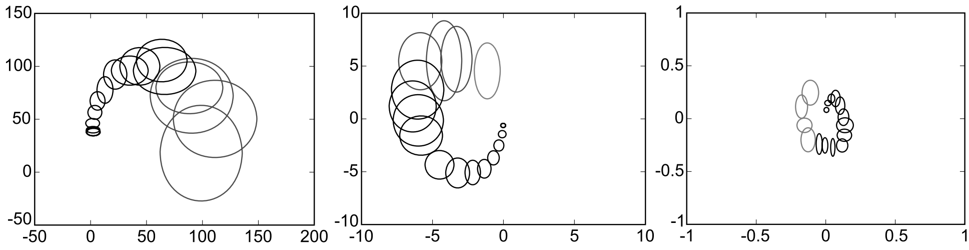

Figure 5. Output probability density functions for HMMs corresponding to motion 'place-on'. The distributions for $\mathbf{x}$, $\dot{\mathbf{x}}$ and $\ddot{\mathbf{x}}$ are shown in the left-hand, middle and right-hand panels, respectively. The vertical and horizontal axes show the C_1 coordinate system (pixel).

log-likelihood of an n-state HMM given a validation set. The trajectories for the motion 'place-on' in C_1 were used as training samples.

Figure 6 illustrates that the validation set likelihood increased with the increase in the number of training samples. Specifically, the likelihood was very small when the sample number was two, while the likelihood converged after six samples were obtained. This indicates that the learned models suffered from over-fitting in the early stage of learning, but appropriate models were finally obtained.

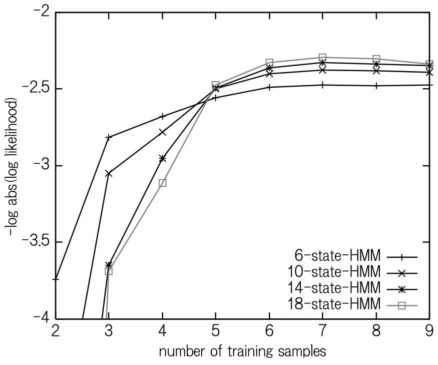

Figure 6. Validation set likelihood of n-state HMMs.

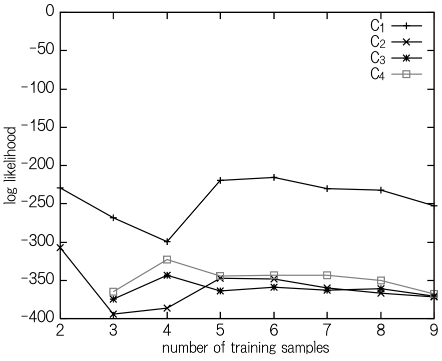

Figure 7. Evolution of training set likelihood.

From Fig. 6, the likelihood of a six-state HMM can be seen to be maximum when the number of samples is less than five; otherwise that of an 18-state HMM is maximum. Therefore, the 18-state HMM was selected as a result of cross-validation when TA-9 was used.

Figure 7 shows the evolution of the training set likelihood. In Fig. 7, the average log-likelihood of intrinsic coordinate systems for the motion ‘place-on’ is plotted. The likelihood of C_3 and C_4 was not obtained when TA-2, which contained two samples, was used since the likelihood was too small due to lack of data. The average log-likelihood of each coordinate system was re-estimated by using L samples, so the likelihood of C_1 was not identical with the data in Fig. 6.

Figure 7 shows that the likelihood decreases with an increase in the number of training samples. This can be explained by a general fact that estimated probabilistic models suffer from over-fitting in the early phase of statistical learning. Therefore, the training set likelihood of an n-state HMM given a small number of samples was larger than the likelihood of the HMM given more samples. From Fig. 7, the likelihood of C_1 increased when the sample number was five. This indicates that the 18-state HMM, which fits the observations better than others, was trained with a sufficient number of samples after five samples (cf., Fig. 6) were obtained.

Figure 8 compares the log-likelihood of intrinsic coordinate systems under the condition that TA-9 was used for training. Figure 8 shows that appropriate coordinate systems were selected for the motions. Table 2 summarizes the relationship of motions and the corresponding types of their intrinsic coordinate systems. Table 2 reveals that C_4 was not selected for any motion, indicating that the learning trajectories of these motions in the camera coordinate system was inappropriate.

The results regarding the parameter estimation described in the Appendix are shown in Fig. 9. Figure 9a shows the evolution of weight w_{l,r_l} for the C_2-type intrinsic coordinate system. In this case, three candidate landmarks were in the sample

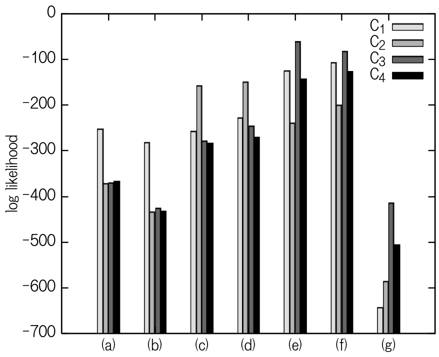

Figure 8. Average log-likelihood of intrinsic coordinate systems: (a) 'place-on', (b) 'jump-over', (c) 'move-closer', (d) 'move-away', (e) 'raise', (f) 'put-down', (g) 'rotate'.

Table 2.
Selected type of intrinsic coordinate system

Type	Motion labels
C_1	jump-over, place-on
C_2	move-away, move-closer
C_3	put-down, raise, rotate
C_4	NA

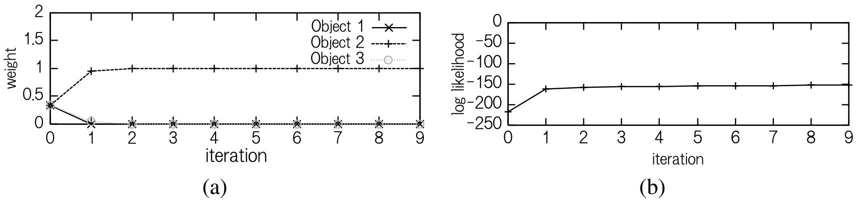

Figure 9. (a) Evolution of w_{l,r_l} against iteration. (b) Evolution of the average log-likelihood.

and w_{l,r_l} converged after three iterations. w_{l,r_l} was confirmed to be converged after less than three iterations for all training samples. Figure 9b reveals how the average log-likelihood increased with the increase in the number of iterations. These two results clearly indicate that the proposed method successfully estimated the correct object as the landmark.

5. Experiment 2: Motion Generation

5.1. Experimental Setup

The objective of the motion generation experiments was to show how the generated trajectories improved when the size of training set increased.

To avoid generating motions in any of trained object placement, another set was obtained from Subject A. The test set RA-5 consisted of samples of seven kinds of motions and each motion has five trajectories. To investigate a generation error decrease with the increase in training set size, trajectories were generated from the models obtained from training sets TA-2 to TA-9. A trajectory performed by Subject A, $\mathcal{Y}$, and generated trajectory, $\hat{\mathcal{Y}}$, can be compared since both $\mathcal{Y}$ and $\hat{\mathcal{Y}}$ were performed/generated given $(r_{\text{traj}}, v_i, r)$.

To evaluate generation error $D(\mathcal{Y}, \hat{\mathcal{Y}})$, Euclidean distance was used since it is a widely used criterion for time series data [19]. Let $\mathbf{x}(t)$ and $\hat{\mathbf{x}}(t)$ denote the position sequences of $\mathcal{Y}$ and $\hat{\mathcal{Y}}$, respectively. The generation error $D(\mathcal{Y}, \hat{\mathcal{Y}})$ was defined as:

$$D(\mathcal{Y}, \hat{\mathcal{Y}}) = \frac{1}{T+1} \sum_{t=0}^{T} \sqrt{|\mathbf{x}(t) - \hat{\mathbf{x}}(t)|^2}, \tag{14}$$

where $D(\mathcal{Y}, \hat{\mathcal{Y}})$ is normalized by the frame length $T+1$ and $\hat{\mathbf{x}}(t)$ is resampled so that $\hat{\mathbf{x}}(t)$ contained the same number of points with $\mathbf{x}(t)$. Note that $D(\mathcal{Y}, \hat{\mathcal{Y}})$ evaluates similarity of two position sequences but does not explicitly evaluate similarity in terms of velocity and acceleration. The same is equally true of most standard techniques for measuring similarity between two time series such as simple dynamic time warping.

The Speech Signal Processing Toolkit was used for generating trajectories.

5.2. Results

The qualitative results regarding the relationship between the generated trajectories and the training set size are shown in Fig. 10. The objects were placed as in Fig. 10, and trajectories for placing Object 3 on Object 1 were generated. In Fig. 10, 'Subject A' represents the trajectory performed by Subject A under the same condition and 'TA-*n*' represents trajectories generated from models trained with *n* samples.

Figure 10 shows that the last position of TA-2's trajectory was not close to the position where it was supposed to be; however, the last positions of TA-8's and Subject A's were almost the same. This is explained by the variances of the last states of the trained models. Indeed, the variances of vertical positions in the models

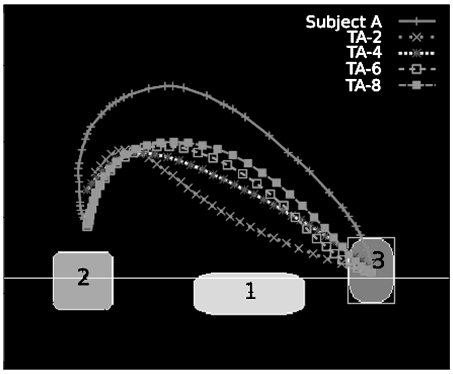

Figure 10. Example generated trajectories from models trained by two, four, six and eight samples.

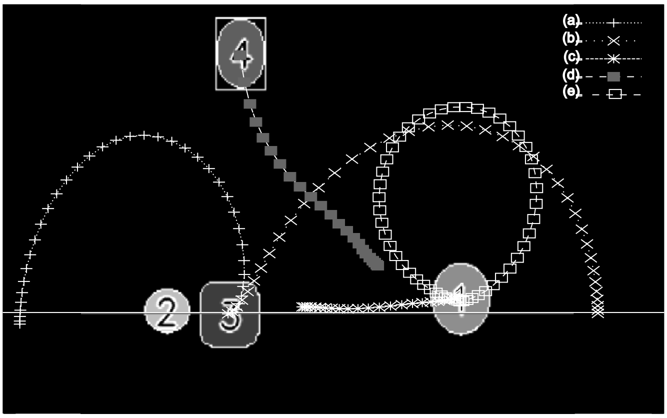

Figure 11. Example generated trajectories for various combinations of relevant objects. The following sets of [motion label, trajector ID, landmark ID] are used: (a) [jump-over, 3, 2], (b) [jump-over, 3, 1], (c) [move-closer, 4, 1], (d) [move-closer, 1, 3] and (e) [rotate, 1].

trained by TA-2, TA-4, TA-6 and TA-8 were 26.6, 23.5, 5.41 and 7.44, respectively. This clearly supports the result shown in Fig. 10.

From Fig. 10, we can see that there were differences in the height dimension between the test sample (Subject A's trajectory) and the generated trajectories. This can be explained by the fact that the subject's trajectory was affected by Object 1; however, the generated trajectories were not. In other words, the subject performed object manipulation, avoiding object collision, while the proposed method simply generated trajectories from probabilistic models representing appearance-level object manipulation. Figure 11 also qualitatively illustrates that appropriate trajectories were generated for various combination of relevant objects. For example, trajectories (a) and (b) were originally generated from the same HMM, but they ended in opposite directions.

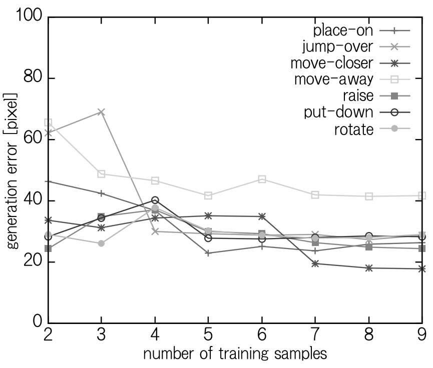

Figure 12. Evolution of generation error $D(\mathcal{Y}, \hat{\mathcal{Y}})$ plotted against the number of training samples.

Figure 12 shows the quantitative evaluation of the proposed method. In Fig. 12, $D(\mathcal{Y}, \hat{\mathcal{Y}})$ averaged by the number of trajectories is plotted against the number of training samples. Figure 12 reveals that $D(\mathcal{Y}, \hat{\mathcal{Y}})$ for the motions 'place-on', 'jump-over', 'move-closer' and 'move-away' decreased with the increase in the number of training samples. The generation error converged after seven samples were used for training. This indicates that seven samples were sufficient for the proposed method to learn motions performed by Subject A in the experiments.

Figure 12 illustrates that the 'move-away' trajectory has larger error than other trajectories. This can be explained as follows. In the training and test set, each trajectory shown by the subject started from the trajector's initial position and ended at various points, increasing the distance from the landmark. Therefore, the difference between the generated and Subject A's trajectory became large, particularly at the final position. The large variances in the trained OPDFs of the 'move-away' HMM also support this explanation. On the other hand, the error of 'move-closer' was smallest among other motions when the sample number was nine. This indicates that the OPDFs of the 'move-closer' HMM had smaller variances.

Figure 13 shows sequential photographs of the manipulator executing trajectories generated by the proposed method. The grasping motions were not obtained through learning, but implemented by the designer.

6. Experiment 3: Motion Recognition

6.1. Experimental Setup

The objective of the motion recognition experiments was to investigate the recognition accuracy of the proposed method. The probabilistic models trained by TA-9 were tested with RA-5 (see Section 5.1). In order to evaluate the recognition accuracy for other persons' motions, we obtained two more test sets, RB-5 and RC-5, from Subjects B and C in the same manner as RA-5.

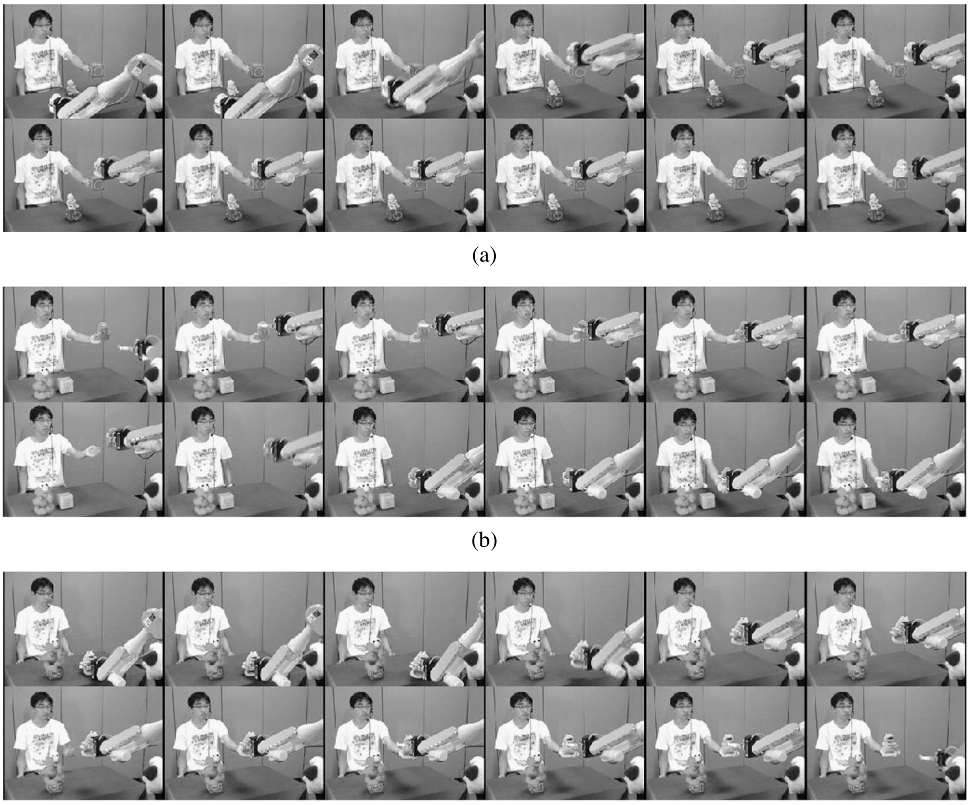

Figure 13. Sequential photographs of the manipulator executing manipulation trajectories. Grasping behavior was implemented by the designer. (a) ‘Place-on’, (b) ‘move-closer’ and (c) ‘raise’.

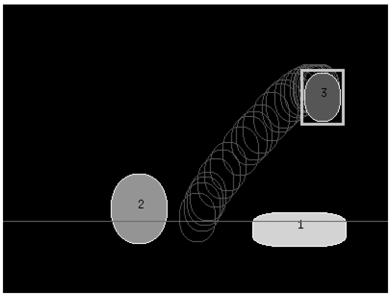

Figure 14. Example of motion recognition results. (*1) [move-away, 2]: −19.59. (2) [move-closer, 1]: −30.59. (3) [jump-over, 1]: −31.62.

6.2. Results

First, a qualitative result is shown in Fig. 14. Figure 14 shows the trajectory that stands for moving Object 3 away from Object 2. The top three recognition results

Table 3.
Number of correct recognition results and accuracy

Training set:	TA-9	TA-9	TA-9
Test set:	RA-5	RB-5	RC-5
place-on	5	5	5
jump-over	5	5	4
move-closer	5	4	3
move-away	5	5	5
raise	5	5	5
put-down	4	3	3
rotate	5	4	5
Total	34 (97%)	31 (89%)	30 (86%)

and their scores (log-likelihood) are shown in the legend. Each recognition result stands for the motion label and the index of the landmark. The result with an asterisk represents the correct recognition. Figure 14 shows that successful recognition was achieved.

Next, the quantitative results are shown in Table 3. The recognition accuracies of RA-5, RB-5 and RC-5 were 97, 89 and 86%, respectively, for an average recognition accuracy of 90%. The average number of objects contained in a test sample was 3.11 and the average chance performance was 3.03%.

7. Discussion

Motion learning, if achieved, can be a considerable step towards language acquisition by robots. In this paper, motions and objects were specified in the form of indices since the focus was on the evaluation of motion learning; however, motions can be mapped with utterances if the proposed method is combined with the learning of phoneme sequences. In fact, the proposed method has been integrated in a probabilistic framework with other modules for learning phoneme sequences, visual features, simple grammar and motion–object relationships [6]. Furthermore, we are currently investigating the generation and recognition of sequential motions by combining reference-point-dependent HMMs [20, 21].

There have been arguments on a number of key issues in imitation learning [1, 4]. Among them, what to imitate and how to imitate are two fundamental questions. The first issue is on determining what perceptual aspects are relevant to a task when attempting to imitate a human [1]. The proposed method focused on the estimation of relevant objects in object-manipulation tasks based on the idea that references of motions performed by humans, were implicit. On the other hand, the correspondence problem [22], or the 'how to imitate' question, poses a problem when a user is trying to transfer skills to robots with different bodies.

The proposed method did not deal with the correspondence problem since the learning was done in task space since the main focus is on the estimation of relevant objects. Unlike most related work in the area of imitation learning, the joint space information is not used in this study. Although using joint space information might enable the method to learn various motions, our approach has a practical advantage in that no motion capture system is needed. With this feature, the proposed method can be introduced to service robots even if a motion capture system is unavailable or the estimation of the user's motion from visual information is unreliable.

For future work, it would be an interesting research topic to improve the proposed method so that it can deal with the correspondence problem for learning more various motions such as ego-centric motions [4]. Some studies have attempted to solve the correspondence problem by extracting important features from task and joint spaces [10].

The basic idea of clustering trajectories by estimating reference points and intrinsic coordinate systems is not dependent on specific methods for modeling time-series data, so other methods such as linear autoregressive models, recurrent neural networks or probabilistic models could be used instead of HMMs. If such a method were not based on a probabilistic framework, the likelihood-based criterion used in this study could be replaced with other criteria, such as prediction errors.

8. Related Work

This section presents related work on imitation learning from a broader perspective, so here the topic is not limited to object manipulation. Early studies on imitation learning in robotics dealt with an assembly task and a kendama (ball-in-cup) task [3, 23]. Recent studies and terminology on imitation learning were presented in Ref. [1]. In Ref. [4], machine learning methods used in imitation learning were reviewed, and different approaches taken within the computer vision, robotics and artificial intelligence communities were analyzed. Reference [2] is an extensive overview of neurocomputational approaches to motor learning by imitation.

Tani *et al.* took a connectionist approach to imitation learning [24]. They used the recurrent neural network with parametric bias (RNNPB) to train the controller of a 4-d.o.f. robot arm. Reference [25] presents an application of RNNPB to the problem of handling many-to-many relationships between motion sequences and linguistic sequences. Okada *et al.* also stressed the importance of self-organizing characteristics for dynamical systems [26]. The recognition and generation of periodical motions are possible with their method.

Synthesizing human-like animation and programming humanoid robots to perform new tasks are attractive applications of methods based on the learning-from-observation framework [1]. Inamura *et al.* used HMMs to model motion patterns such as 'swing' and 'walking' [12]. The trained HMMs were mapped in a space by applying multidimensional scaling to Kullback–Leibler divergences between the parameters of the HMMs. Novel motions were generated by the interpolation and extrapolation of the HMMs in the mapped space [27]. A hidden semi-Markov model, which is an HMM with state duration probability distributions, was used for learning walking patterns in Ref. [28]. In Ref. [29], a Gaussian process dynamical model (GPDM) was proposed and applied to human motion capture data. A GPDM comprises of a low-dimensional latent space with associated dynamics and a map from the latent space to an observation space.

These studies focused on the learning motions in a fixed coordinate system. In contrast, the proposed method in this study is able to estimate the optimal intrinsic coordinate system to generalize motion trajectories.

9. Conclusions

This paper presented a method for learning object-manipulation motions from multiple demonstrations by humans. Conventional methods for learning object manipulation from demonstration had trouble dealing with motions that did and did not depend on a landmark with the same learning framework. In contrast, one of the contributions of the paper was to provide an algorithm for estimating reference points to overcome such a limitation.

The proposed method was applied to the learning of object manipulation and evaluated its generation errors and recognition accuracy. While the results in this paper are promising, the learned motions were limited to an appearance level. In the future the application of machine learning to the problem of understanding the intent of an action will hopefully be explored, as that is necessary to develop a robot that imitates the goal of an action.

Acknowledgements

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 20500186, 2008, and the National Institute of Informatics.

Note

An earlier version of this work was presented in the *Proceedings of the 2007 Workshop on Multimodal Interfaces in Semantic Interaction* [19].

References

1. C. Breazeal and B. Scassellati, Robots that imitate humans, *Trends Cognit. Sci.* **6**, 481–487 (2002).
2. S. Schaal, A. Ijspeert and A. Billard, Computational approaches to motor learning by imitation, *Phil. Trans. R. Soc. London B* **358**, 537–547 (2003).
3. Y. Kuniyoshi, M. Inaba and H. Inoue, Learning by watching: extracting reusable task knowledge from visual observation of human performance, *IEEE Trans. Robotics Automat.* **10**, 799–822 (1994).
4. V. Krüger, D. Kragic, A. Ude and C. Geib, The meaning of action: a review on action recognition and mapping, *Adv. Robotics* **21**, 1473–1501 (2007).

5. D. Roy, Grounding words in perception and action: computational insights, *Trends Cognit. Sci.* **9**, 389–396 (2005).
6. N. Iwahashi, Robots that learn language: developmental approach to human–machine conversations, in: *Human–Robot Interaction*, N. Sanker (Ed.), pp. 95–118. I-Tech Education and Publishing, Vienna (2007).
7. T. Regier, *The Human Semantic Potential: Spatial Language and Constrained Connectionism*. Bradford Books/MIT Press, Cambridge, MA (1996).
8. K. Ogawara, J. Takamatsu, H. Kimura and K. Ikeuchi, Generation of a task model by integrating multiple observations of human demonstrations, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington, DC, pp. 1545–1550 (2002).
9. K. Ogawara, J. Takamatsu, H. Kimura and K. Ikeuchi, Modeling manipulation interactions by hidden Markov models, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Lausanne, pp. 1096–1101 (2002).
10. A. Billard, S. Calinon and F. Guenter, Discriminative and adaptive imitation in uni-manual and bi-manual tasks, *Robotics Autonomous Syst.* **54**, 370–384 (2006).
11. S. Calinon and A. Billard, A probabilistic programming by demonstration framework handling constraints in joint space and task space, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nice, pp. 367–372 (2008).
12. T. Inamura, I. Toshima, H. Tanie and Y. Nakamura, Embodied symbol emergence based on mimesis theory, *Int. J. Robotics Res.* **23**, 363–377 (2004).
13. K. Tokuda, T. Kobayashi and S. Imai, Speech parameter generation from HMM using dynamic features, in: *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, MI, pp. 660–663 (1995).
14. R. W. Langacker, *Foundations of Cognitive Grammar: Theoretical Prerequisites*. Stanford University Press, Stanford, CA (1987).
15. L. A. Carlson-Radvansky and D. E. Irwin, Frames of reference in vision and language: where is above?, *Cognition* **46**, 223–244 (1993).
16. A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum likelihood from incomplete data *via* the EM algorithm, *J. R. Stat. Soc. B* **39**, 1–38 (1977).
17. C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, Berlin (2006).
18. HTK web site, The Hidden Markov Model Toolkit, http://htk.eng.cam.ac.uk.
19. E. Keogh and S. Kasetty, On the need for time series data mining benchmarks: a survey and empirical demonstration, *Data Mining Knowledge Discov.* **7**, 349–371 (2003).
20. K. Sugiura and N. Iwahashi, Learning object-manipulation verbs for human–robot communication, in: *Proc. Workshop on Multimodal Interfaces in Semantic Interaction*, Nagoya, pp. 32–38 (2007).
21. K. Sugiura and N. Iwahashi, Motion recognition and generation by combining reference-point-dependent probabilistic models, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nice, pp. 852–857 (2008).
22. C. Nehaniv and K. Dautenhahn, The correspondence problem, in: *Imitation in Animals and Artifacts*, pp. 41–61. MIT Press, Cambridge, MA (2002).
23. H. Miyamoto, S. Schaal, F. Gandolfo, H. Gomi, Y. Koike, R. Osu, E. Nakano, Y. Wada and M. Kawato, A kendama learning robot based on bi-directional theory, *Neural Netw.* **9**, 1281–1302 (1996).
24. J. Tani and M. Ito, Self-organization of behavioral primitives as multiple attractor dynamics: a robot experiment, *IEEE Trans. Syst. Man Cybernet. A* **33**, 481–488 (2003).

25. T. Ogata, M. Murase, J. Tani, K. Komatani and H. G. Okuno, Two-way translation of compound sentences and arm motions by recurrent neural networks, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and System*, San Diego, CA, pp. 1858–1863 (2007).
26. M. Okada, D. Nakamura and Y. Nakamura, Self-organizing symbol acquisition and motion generation based on dynamics-based information processing system, in: *Proc. Int. Workshop on Man–Machine Symbiotic Systems*, Kyoto, pp. 219–229 (2004).
27. T. Inamura and T. Shibata, Geometric proto-symbol manipulation towards language-based motion pattern synthesis and recognition, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nice, pp. 334–339 (2008).
28. N. Niwase, J. Yamagishi and T. Kobayashi, Human walking motion synthesis with desired pace and stride length based on HSMM, *IEICE Trans. Inform. Syst.* **88**, 2492–2499 (2005).
29. J. M. Wang, D. J. Fleet and A. Hertzmann, Gaussian process dynamical models, *Adv. Neural Inform. Process. Syst.* **18**, 1441–1448 (2006).
30. L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. IEEE* **77**, 257–286 (1989).

Appendix: Parameter Estimation of Reference-Point-Dependent HMMs

This appendix explains a parameter estimation method to solve (8). The superscript k is omitted for the sake of simplicity since k is fixed in the following equations.

First, (8) is simplified before applying an EM algorithm. Since $w_{l,r_l} = P(r_l|\mathbf{w})$, this results in:

$$\begin{aligned} w_{l,r_l} P\left({}^{C_k(\mathbf{x}^{r_l})}\mathcal{Y}_l; \lambda\right) &= P(r_l|\mathbf{w}) P(\mathcal{Y}_l|r_l, k, \lambda) \\ &= P(\mathcal{Y}_l, r_l|k, \lambda, \mathbf{w}) \\ &= P(\mathcal{Y}_l, r_l|k, \Phi), \end{aligned} \tag{A.1}$$

where:

$$\Phi \triangleq (\lambda, \mathbf{w}). \tag{A.2}$$

Applying (A.1) and (A.2), (8) can be rewritten as:

$$\begin{aligned} \hat{\Phi} &= \arg\max_{\Phi} \sum_{l=1}^{L} \log\left[\sum_{r_l=1}^{|\mathbf{R}_l|} P(\mathcal{Y}_l, r_l|k, \Phi)\right] \\ &= \arg\max_{\Phi} \sum_{l=1}^{L} \log P(\mathcal{Y}_l|k, \Phi). \end{aligned} \tag{A.3}$$

Here, an HMM is employed as the probabilistic model. Let $\boldsymbol{\pi} = \{\pi_i\}$ denote the initial probability vector, $A = \{a_{ij}\}$ denote the state transition probability matrix, $\mathbf{b} = \{b_j(\mathbf{y})\}$ denote the OPDFs and $\mathbf{q}_l = \{q_{l,t}\}$ denote the state transition vector of the lth learning data. When an EM algorithm is applied by regarding r_l and $\mathbf{q}_l$ as hidden parameters, the following auxiliary function $Q(\Phi, \Phi')$ is derived:

$$Q(\Phi, \Phi') = \sum_{l=1}^{L} \sum_{r_l=1}^{|\mathbf{R}_l|} \sum_{\mathbf{q}_l} P\left(r_l, \mathbf{q}_l|\mathcal{Y}_l, k, \Phi'\right) \log P(\mathcal{Y}_l, r_l, \mathbf{q}_l|k, \Phi). \tag{A.4}$$

Each term in the above equation is represented with HMM parameters as:

$$P(r_l, \mathbf{q}_l | \mathcal{Y}_l, k, \Phi') = \frac{w'_{l,r_l}}{P_l} P\left({}^{C_k(\mathbf{x}^{r_l})}\mathcal{Y}_l, \mathbf{q}_l | \lambda'\right) \tag{A.5}$$

$$\begin{aligned}&\log P(\mathcal{Y}_l, r_l, \mathbf{q}_l | k, \Phi) \\ &\quad = \log w_{l,r_l} + \log \pi_{q_{l,0}} + \sum_{t=1}^{T_l} \log a_{q_{l,t} q_{l,t+1}} + \sum_{t=1}^{T_l} \log b_{q_{l,t+1}} \mathbf{z}_l(t),\end{aligned} \tag{A.6}$$

where $\mathbf{z}_l(t) \triangleq {}^{C_k(\mathbf{x}^{r_l})}\mathbf{y}_l(t)$.

From (A.4)–(A.6), the following parameter re-estimation equations are derived, which maximize $Q(\Phi, \Phi')$ in terms of Φ:

$$\bar{w}_{l,r_l} = \frac{w_{l,r_l}}{P_l} \sum_{i=1}^{N} \alpha_{l,r_l}(T_l, i) \tag{A.7}$$

$$\bar{\pi}_i = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{P_l} \sum_{r_l=1}^{|\mathbf{R}_l|} w_{l,r_l} \alpha_{l,r_l}(0, i) \beta_{l,r_l}(0, i) \tag{A.8}$$

$$\bar{a}_{ij} = \frac{\sum_{l=1}^{L} 1/P_l \sum_{r_l=1}^{|\mathbf{R}_l|} w_{l,r_l} \sum_{t=1}^{T_l} \alpha_{l,r_l}(t, i) a_{ij} b_j \mathbf{z}_l(t+1) \beta_{l,r_l}(t+1, j)}{\sum_{l=1}^{L} 1/P_l \sum_{r_l=1}^{|\mathbf{R}_l|} w_{l,r_l} \sum_{t=1}^{T_l} \alpha_{l,r_l}(t, i) \beta_{l,r_l}(t, i)} \tag{A.9}$$

$$P_l \triangleq P(\mathcal{Y}_l | k, \Phi), \tag{A.10}$$

where N, $\alpha_{l,r_l}(t, i)$ and $\beta_{l,r_l}(t, i)$ denote the number of the states, and the forward and backward parameters [30] for the HMM, respectively.

When b_j is single Gaussian, the following re-estimation equations of the mean vector $\boldsymbol{\mu}_j$ and variance–covariance matrix $\boldsymbol{\Sigma}_j$ are obtained:

$$\bar{\boldsymbol{\mu}}_j = \frac{\sum_{l=1}^{L} \sum_{r_l=1}^{|\mathbf{R}_l|} \sum_{t=1}^{T_l} \gamma_l(t, j) \mathbf{z}_l(t)}{\sum_{l=1}^{L} \sum_{r_l=1}^{|\mathbf{R}_l|} \sum_{t=1}^{T_l} \gamma_l(t, j)} \tag{A.11}$$

$$\bar{\boldsymbol{\Sigma}}_j = \frac{\sum_{l=1}^{L} \sum_{r_l=1}^{|\mathbf{R}_l|} \sum_{t=1}^{T_l} \gamma_l(t, j) [\mathbf{z}_l(t) - \boldsymbol{\mu}_j][\mathbf{z}_l(t) - \boldsymbol{\mu}_j]^\top}{\sum_{l=1}^{L} \sum_{r_l=1}^{|\mathbf{R}_l|} \sum_{t=1}^{T_l} \gamma_l(t, j)}, \tag{A.12}$$

where:

$$\gamma_l(t, j) \triangleq \frac{w_{l,r_l}}{P_l} \alpha_{l,r_l}(t, j) \beta_{l,r_l}(t, j). \tag{A.13}$$

About the Authors

Komei Sugiura is an Expert Researcher at the Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology (NICT). He received his BE degree in Electrical and Electronic Engineering, and MS and PhD degrees in Informatics from Kyoto University, in 2002, 2004 and 2007, respectively. From 2006 to 2008, he was a Research Fellow, Japan Society for the Promotion of Science, and he has been with NICT since 2008. His research interests include robot language acquisition, spoken dialogue systems, machine learning, sensor evolution and service robots.

Naoto Iwahashi received the BE degree in engineering from Keio University, Yokohama, Japan, in 1985. He received the PhD degree in Engineering from Tokyo Institute of Technology, in 2001. In 1985, he joined Sony Corp., Tokyo, Japan. From 1990 to 1993, he was at ATR Interpreting Telephony Research Laboratories, Kyoto, Japan. From 1998 to 2003, he was with Sony Computer Science Laboratories Inc., Tokyo, Japan. In 2003, he joined ATR Spoken Language Communication Research Laboratories. In 2006, he joined the National Institute of Information and Communications Technology. His research areas include interactive speech system, language acquisition and human–robot interaction.

Hideki Kashioka received his PhD in Information Science from Osaka University, in 1993. From 1993 to 2009, he worked for ATR Spoken Language Translation Research Laboratories. From 2006, he has worked for the National Institute of Information and Communications Technology (NICT). He is currently the Research Manager of the Spoken Language Communication Group at the Knowledge Creating Communication Research Center, NICT. He is also Visiting Associate Professor of the Graduate School of Information Science at the Nara Institute of Science and Technology, from 1999.

Satoshi Nakamura received his BS from Kyoto Institute of Technology, in 1981, and PhD from Kyoto University, in 1992. He was an Associate Professor of the Graduate School of Information Science at Nara Institute of Science and Technology, in 1994–2000. He was Director of an ATR Spoken Language Communication Research Laboratories, in 2000–2008. He is an ATR Fellow. He launched the world's first network-based commercial speech-to-speech translation service for 3G mobile phones, in 2007. He is currently the Director General of Keihanna Research Laboratories and the Executive Director of the Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology, Japan. He received the Yamashita Research Award, Kiyasu Award from the Information Processing Society of Japan, Telecom System Award, AAMT Nagao Award, Docomo Mobile Science Award, in 2007, Telecom System Award, ASJ Award for Distinguished Achievements in Acoustics, and the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology.