2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Motion Generation by Reference-Point-Dependent Trajectory HMMs

Komei Sugiura, Naoto Iwahashi and Hideki Kashioka

Abstract— **This paper presents an imitation learning method for object manipulation such as rotating an object or placing one object on another. In the proposed method, motions are learned using reference-point-dependent probabilistic models. Trajectory hidden Markov models (HMMs) are used as the probabilistic models so that smooth trajectories can be generated from the HMMs. The method was evaluated in physical experiments in terms of motion generation. In the experiments, a robot learned motions from observation, and it generated motions under different object placement. Experimental results showed that appropriate motions were generated even when the object placement was changed.**

I. INTRODUCTION

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 user motions, obtaining a controller for motions adapted to each home is possible.

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 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 [1]. Regier investigated a model describing the spatial relationship between two objects [2]. 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) [3], [4]. Billard *et al.* developed a method which 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 [5], 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 [6]. 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, learning the "place an object on another" motion requires the position sequences of two objects, while learning the "raise an object" motion does not. Thus, overcoming this limitation enables the learning of various types of object manipulation from natural interactions between the user and robot.

On the other hand, we have investigated imitation learning methods for reference-point-dependent motions [7]–[9]. The number of objects necessary to describe the motion is automatically determined, and the relevant object(s) in each demonstration is estimated as well. In this paper, we present a motion generation method for reference-point-dependent motions. Our key contributions are as follows:

- Motion generation method by reference-point-dependent trajectory HMMs. The HMMs are trained in the *intrinsic coordinate system* (see Section III) and transformed to the world coordinate system. The transformation and motion generation methods are explained in Section IV and Appendix.
- Smooth trajectories are generated by using trajectory HMMs (see Fig. 4). The generated trajectories are evaluated in Section VI.

In the paper, the method is evaluated with a fixed manipulator robot; however, the method is also applicable to wheeled humanoid robots. Indeed, the proposed method was successfully applied to learning to throw an object into another in the RoboCup@Home [10] environment.

II. THE TASK

A. 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 [11]. An example of this is shown in the left-hand figure of Fig. 1. The figure depicts a camera image in which the green puppet (Kermit) is moved along the dotted line. In the example shown in Fig. 1, the trajector is Kermit. On the other hand, there are multiple

Komei Sugiura, Naoto Iwahashi and Hideki Kashioka are with the National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika, Soraku, Kyoto 619-0289, Japan. komei.sugiura@nict.go.jp

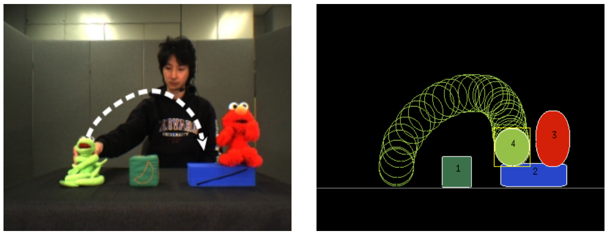

Fig. 1. Left: Example shot of an image stream. Right: Preprocessed visual features obtained from the image stream.

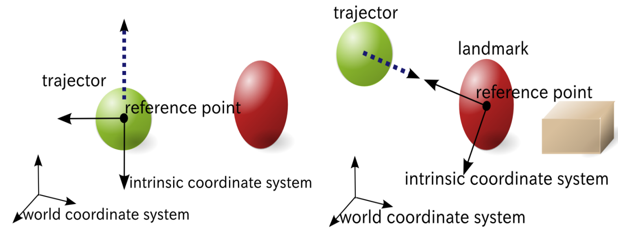

Fig. 2. Relationship among a trajector/landmark, a reference point, and an intrinsic coordinate system. The spheres, ellipsoids, and box represent objects, and the arrows represent the axes of the intrinsic coordinate systems. Left: "raise." Right: "move-closer."

interpretations of the landmark. If the blue 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 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 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 [12] 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. 2). The reference point of "raise" can reasonably be assumed to be the trajector's center of gravity. The intrinsic coordinate system can be a Cartesian coordinate system, as shown in the left-hand figure. 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.

B. Hardware Platform

The experiments were conducted with the platform shown in Fig. 3. 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 $\times$ 240 pixels. The left-hand figure of Fig. 1 shows an example shot of an image stream, and the right-hand figure shows its internal representation. The transformation matrix between the camera coordinate system and the world coordinate system was fixed since the

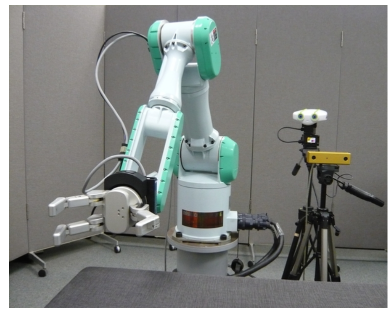

Fig. 3. Hardware platform used in the experiments.

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 degrees of freedom (DOFs). The manipulator was equipped with a BarrettHand, a four-DOF 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.

III. Motion Learning by Reference-Point-Dependent Trajectory HMMs

A. Modeling Trajectory by Delta Parameters

Let $\mathcal{V}$ denote a single training sample for a type of motion such as "raise" or "place-on." $\mathcal{V}$ consists of the trajectory of trajector, Ξ, and a set of positions of the static objects, $\boldsymbol{O}_S$, as follows:

$$\mathcal{V} = (\Xi, \boldsymbol{O}_S), \tag{1}$$

$$\Xi = \left[\boldsymbol{\xi}_1^\top, \boldsymbol{\xi}_2^\top, ..., \boldsymbol{\xi}_T^\top,\right]^\top, \tag{2}$$

$$\boldsymbol{\xi}_t = \left[\boldsymbol{x}_t^\top, \Delta^{(1)}\boldsymbol{x}_t^\top, \Delta^{(2)}\boldsymbol{x}_t^\top\right]^\top, \tag{3}$$

where $\boldsymbol{x}_t$ denotes the position vector at time t. The dimension of $\boldsymbol{x}_t$ is denoted by m. We assume that $m = 2$; however, the method can be applied to three-dimensional space. $\Delta^{(k)}\boldsymbol{x}_t$ $(k = 0, 1, 2)$ is called a delta parameter in the speech processing community. Delta parameters, which are used for generating smooth trajectories, are defined as:

$$\Delta^{(k)}\boldsymbol{x}_t = \sum_{\tau=-L^{(k)}}^{L^{(k)}} w^{(k)}(\tau)\boldsymbol{x}_{t+\tau}, \qquad (k = 0, 1, 2), \tag{4}$$

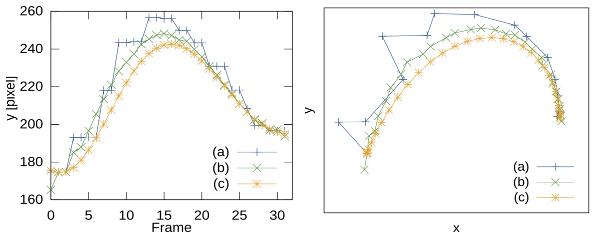

Fig. 4. Examples of generated trajectory for motion "place-on" using (a) $\boldsymbol{x}_t$, (b) $(\boldsymbol{x}_t, \Delta^{(1)}\boldsymbol{x}_t)$, and (c) $(\boldsymbol{x}_t, \Delta^{(1)}\boldsymbol{x}_t, \Delta^{(2)}\boldsymbol{x}_t)$, Left: t-y plot. Right:x-y plot.

where $L^{(0)} = 0$ and $w^{(0)}(0) = 1$. We set $L^{(k)}$ as $L^{(1)} = L^{(2)} = 1$. $w^{(k)}(\tau)$ is set as $w^{(1)}(\tau) = \frac{\tau}{2}$ and $w^{(2)}(\tau) = \frac{3}{4}\tau^2 - \frac{1}{2}$, and thus we obtain

$$\Delta^{(1)}\boldsymbol{x}_t = -\frac{1}{2}\boldsymbol{x}_{t-1} + \frac{1}{2}\boldsymbol{x}_{t+1}, \tag{5}$$

$$\Delta^{(2)}\boldsymbol{x}_t = \frac{1}{4}\boldsymbol{x}_{t-1} - \frac{1}{2}\boldsymbol{x}_t + \frac{1}{4}\boldsymbol{x}_{t+1}. \tag{6}$$

The effect of the delta parameters is shown in Fig. 4. Smooth trajectories are generated with them.

A left-to-right HMM is used for modeling trajectory Ξ. Let λ denote the parameters of the N-state HMM. λ consists of initial state probability $\boldsymbol{\pi} = \{\pi_i\}$, state transition probability A, and output probability density functions (OPDFs) $\boldsymbol{b} = \{b_i(\cdot)\}$. A single-mixture Gaussian distribution is used for the OPDF.

B. Intrinsic Coordinate System Types

We assume that there are several types of intrinsic coordinate systems, and these are given by the designer. The type of intrinsic coordinate system is denoted by k. 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 when the x coordinate of $\boldsymbol{x}_1$ is negative after translation.
- C_2 An orthogonal coordinate system with its origin at the landmark position. The direction of the x axis is from the landmark position toward $\boldsymbol{x}_1$.
- C_3 A translated camera coordinate system with its origin at $\boldsymbol{x}_1$.
- C_4 A translated camera coordinate system with its origin at $\boldsymbol{x}_{\text{center}}$.

In C_3- or C_4-type intrinsic coordinate systems, the reference point does not have to be estimated since it is fixed to $\boldsymbol{x}_1$ or $\boldsymbol{x}_{\text{center}}$.

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

$$\boldsymbol{O}_R = \{\boldsymbol{O}_S, \boldsymbol{x}(0), \boldsymbol{x}_{\text{center}}\} \triangleq \left\{\boldsymbol{x}^r \middle| r = 1, 2, ..., |\boldsymbol{O}_R|\right\}, \tag{7}$$

where $|\boldsymbol{O}_R|$ denotes the number of elements contained in $\boldsymbol{O}_R$.

C. Training Reference-Point-Dependent Trajectory HMMs

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

$$(\hat{\lambda}, \hat{k}, \hat{\boldsymbol{r}}) = \underset{\lambda,k,\boldsymbol{r}}{\operatorname{argmax}} \sum_{l=1}^{L} \log P({}^{C_k(\boldsymbol{x}^r)}\Xi_l; {}^{C_k(\boldsymbol{x}^r)}\lambda), \tag{8}$$

where Ξ_l represents the trajectory in the lth training sample. The solution to Equation (8) is explained in [9] in detail.

To estimate the optimal number of states of HMMs, we use leave-one-out cross-validation [13]. In the first stage of the cross-validation process, training samples were divided into *real* training set ($L-1$ samples) and validation set (one sample).

We obtain state sequence $\bar{\boldsymbol{q}}$ by averaging the state durations:

$$\begin{aligned} \bar{\boldsymbol{q}} &= (q_{\text{init}}, q_1, q_2, ..., q_{\bar{T}}, q_{\text{final}}) \\ &= (S_{\text{init}}, \underbrace{1, ..., 1}_{\bar{d}_1}, \underbrace{2, ..., 2}_{\bar{d}_2}, ..., \underbrace{N, ..., N}_{\bar{d}_N}, S_{\text{final}}), \end{aligned} \tag{9}$$

where $\bar{d}_i$ denotes the average duration of the ith state, and $\bar{T}$ represents the sum of $\bar{d}_i$. $\bar{\boldsymbol{q}}$ is later used for motion generation.

IV. Motion Generation by Reference-Point-Dependent Trajectory HMMs

In this section, we consider the problem of generating trajectories from reference-point-dependent trajectory HMMs. We obtain the following trajectory that maximizes the probability of ${}^W\hat{\Xi}$ conditioned by state sequence $\bar{\boldsymbol{q}}$ and HMM parameters ${}^W\lambda$ as follows:

$$ {}^W\hat{\Xi} = \underset{{}^W\Xi}{\operatorname{argmax}}\, P({}^W\Xi |\, \bar{\boldsymbol{q}}, {}^W\lambda) \tag{10}$$

$$ = \underset{{}^W\Xi}{\operatorname{argmax}}\, P({}^W\Xi |\, \bar{\boldsymbol{q}}, {}^C\lambda, \boldsymbol{x}^{r_{\text{traj}}}, \boldsymbol{x}^r), \tag{11}$$

where C and W respectively represent the intrinsic and world coordinate systems, $\boldsymbol{x}^{r_{\text{traj}}}$ denotes the trajector position, and $\boldsymbol{x}^r$ denotes the reference point.

Since HMM parameters λ are trained in intrinsic coordinate system C, we first conduct the transformation from C to W. Let ${}^C\boldsymbol{\mu}^{(k)}(s)$ and ${}^C\Sigma^{(k)}(s)$ denote the mean vector and the covariance matrix of the $k(=0,1,2)$th delta parameters at state s, respectively.

Mean position vector ${}^{C}\boldsymbol{\mu}^{(0)}(s)$ is transformed by the following homogeneous transformation matrix:

$$\begin{bmatrix} {}^{W}\boldsymbol{\mu}^{(0)}(s) \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} R & \boldsymbol{x}^{r_{\text{traj}}} \\ \hline \mathbf{0} & 1 \end{array}\right] \begin{bmatrix} {}^{C}\boldsymbol{\mu}^{(0)}(s) - {}^{C}\boldsymbol{\mu}^{(0)}(1) \\ 1 \end{bmatrix}, \quad (s = 1, 2, ..., N) \tag{12}$$

where R denotes the rotation matrix. R is also used for rotating the other mean vectors and the covariance matrices:

$$ {}^{W}\boldsymbol{\mu}^{(k)}(s) = R\, {}^{C}\boldsymbol{\mu}^{(k)}(s) \qquad (k = 1, 2) \tag{13}$$

$$ {}^{W}\Sigma^{(k)}(s) = R\, {}^{C}\Sigma^{(k)}(s)\, R^{\top} \qquad (k = 0, 1, 2) \tag{14}$$

To generate a trajectory which starts from $\boldsymbol{x}^{r_{\text{traj}}}$, a positive small number $\varepsilon \ll 1$ is set as the position variance at the first state:

$$ {}^{W}\Sigma^{(0)}(1) = \varepsilon I. \tag{15}$$

Thus, we obtained ${}^{W}\lambda$ from ${}^{C}\lambda$. A solution to Equation (10) is explained in Appendix.

V. Experimental Setup

A. Training and Test Sets

In this subsection, we explain the setup of the experiments conducted to obtain trained reference-point-dependent HMMs for our subsequent motion generation experiments.

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 label (verb) such as "raise" or "rotate." The environment is illustrated in the left-hand figure of Fig. 1. A training sample was obtained from the manipulation by preprocessing the recorded image stream of the manipulation (see the right-hand figure of Fig. 1).

All trajectories were obtained from the demonstration by the same subject, Subject A. The following motion labels were used:

> jump-over, move-away, move-closer, place-on, put-down, raise, rotate

The training sets obtained in the experiments are summarized in Table I. 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.

To avoid generating motions in any of the trained object placement, another set was obtained from Subject A. Test set RA-5 consisted of samples of seven kinds of motions, and each motion has five trajectories.

While the number of states is fixed in most imitation learning studies, we select the optimal number of states by cross-validation. We used four kinds of initial HMMs, each of which had $n \in \{6, 10, 14, 18\}$ states. The average computation time for learning a motion in TA-9 was 9.8 seconds with a quad-core 2.66GHz computer.

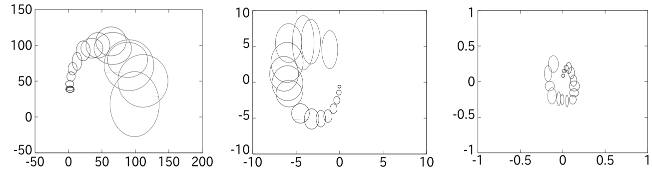

Fig. 5. Output probability density functions for HMM corresponding to motion "place-on." Left-hand, middle, and right-hand figures show the OPDFs of the $k(=0,1,2)$-th delta parameters, respectively. Vertical and horizontal axes show the C_1-coordinate system [pixel].

B. Training Reference-Point-Dependent Trajectory HMMs

Fig. 5 shows the output probability density functions (OPDFs) for an HMM trained with trajectories corresponding to 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 figures of Fig. 5 illustrate the distributions corresponding to the $k(=0,1,2)$-th delta parameters, respectively. The center and semimajor (or semiminor) axis of each ellipse stand for the mean and standard deviation of each distribution. The direction of the state transition is indicated by the darkness of the color. From Fig. 5, we can see that the first and last state for position have large and small variances, respectively. This indicates that the trajectory of "place-on" started from an arbitrary point but converged to a certain point in the end.

Fig. 6 compares the log likelihood of intrinsic coordinate systems under the condition that TA-9 was used for training. The figure shows that appropriate coordinate systems were selected for the motions. Table II summarizes the relationship of the motions and the corresponding types of their intrinsic coordinate systems. Table II reveals that C_4 was not selected for any motion, indicating that the learning trajectories of these motions in the camera coordinate system were inappropriate. Except for "put-down," the optimal state number was 18, and it was 10 for "put-down."

TABLE I
Training and test sets.

Data set	Subject	# of motion labels	# of trajectories per motion label
TA-2	A	7	2
TA-3	A	7	3
⋮	⋮	⋮	⋮
TA-9	A	7	9
RA-5	A	7	5

TABLE II
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	N.A.

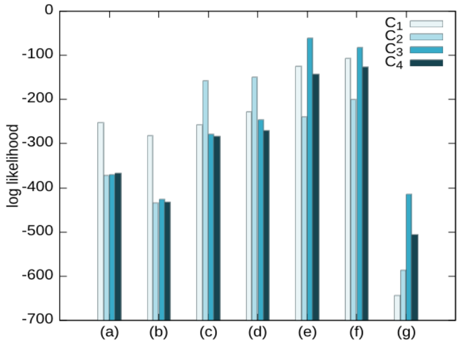

Fig. 6. Average log likelihood of intrinsic coordinate systems. (a)"place-on," (b)"jump-over," (c)"move-closer," (d)"move-away," (e)"raise," (f)"put-down," (g)"rotate."

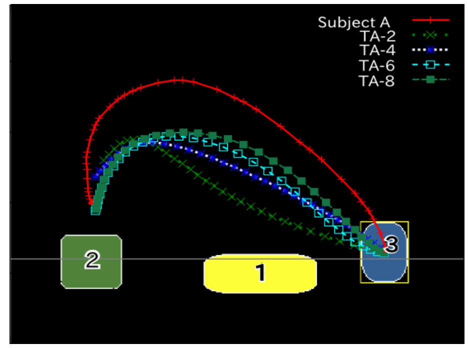

Fig. 7. Generated trajectories from models trained by 2, 4, 6, and 8 samples.

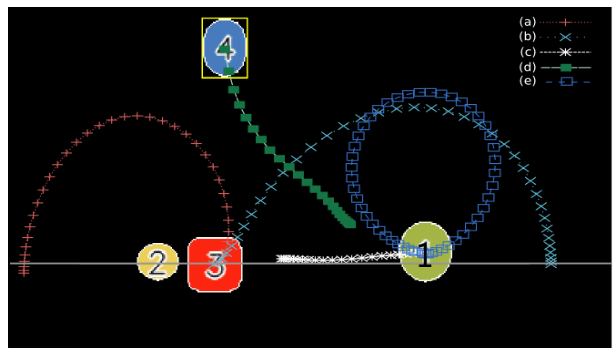

Fig. 8. Generated trajectories for various combinations of relevant objects. The following sets of [motion label, trajector ID, landmark ID] were used: (a) [jump-over, 3, 2], (b) [jump-over, 3, 1], (c) [move-closer, 4, 1], (d) [move-closer, 1, 3], and (e) [rotate, 1].

VI. Experimental Results

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

A. Qualitative Results

The qualitative results regarding the relationship between the generated trajectories and the training set size are shown in Fig. 7. The objects were placed as in the figure, and the trajectories were generated for placing Object 3 on Object 2. "Subject A" represents the trajectory performed by Subject A under the same condition, and "TA-n" represents the trajectories generated from models trained with n samples.

The figure 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 the vertical positions in the models 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. 7.

From Fig. 7, we can see 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 with avoiding object collision, while the proposed method simply generated trajectories from probabilistic models representing appearance-level object manipulation. Fig. 8 also qualitatively illustrates that appropriate trajectories were generated for various combinations of relevant objects. For example, trajectories (a) and (b) were originally generated from the same HMM, but they ended in opposite directions.

B. Quantitative Results

To investigate the 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, Ξ, and generated trajectory $\hat{\Xi}$ can be compared since both Ξ and $\hat{\Xi}$ were performed/generated given $(r_{\text{traj}}, v_i, r)$, where r_{traj} denotes the trajector ID, v_i denotes the motion label, and r denotes the reference point ID.

To evaluate generation error $D(\Xi, \hat{\Xi})$, Euclidean distance was used since it is a widely used criterion for time series data [14]. Let $\boldsymbol{x}_t$ and $\hat{\boldsymbol{x}}_t$ denote the position at frame t in trajectories Ξ and $\hat{\Xi}$, respectively. Generation error $D(\Xi, \hat{\Xi})$ was defined as follows:

$$D(\Xi, \hat{\Xi}) = \frac{1}{T} \sum_{t=1}^{T} \sqrt{|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t|^2}, \tag{16}$$

where $D(\Xi, \hat{\Xi})$ is normalized by frame length T, and $\hat{\boldsymbol{x}}_t$ is resampled so that $\hat{\boldsymbol{x}}_t$ contained the same number of points with $\boldsymbol{x}_t$. Note that $D(\Xi, \hat{\Xi})$ evaluates the similarity of the two position sequences without explicitly evaluating the similarity in terms of the velocity and acceleration[1] .

Fig. 9 shows a quantitative evaluation of the proposed method. In the figure, $D(\Xi, \hat{\Xi})$, which was averaged by the number of trajectories, is plotted against the number of training samples. Fig. 9 reveals that $D(\Xi, \hat{\Xi})$ for 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

[1]The same is equally true of most standard techniques for measuring similarity between two time series such as simple Dynamic Time Warping.

were sufficient for the proposed method to learn motions performed by Subject A in the experiments.

Fig. 9 illustrates that the “move-away” trajectory has larger error than other trajectories for the following reason. In the training and test sets, 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 trajectories increased 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.

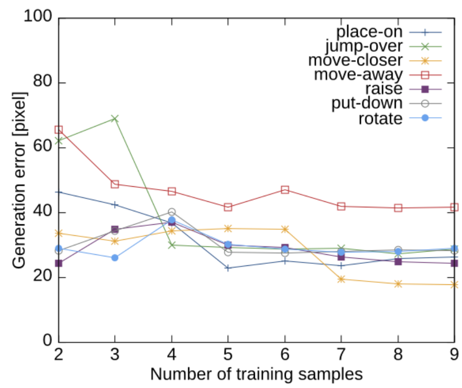

Fig. 9. Evolution of generation error $D(\Xi, \hat{\Xi})$ plotted against number of training samples.

Fig. 10 shows sequential photographs of the manipulator executing trajectories generated by the proposed method. The grasping motions were not obtained through learning but were implemented by the designer.

VII. Discussion

A. Future Work

Motion generation from trajectory HMMs is not limited to left-to-right HMMs. The method explained in Section III and Appendix can be applied to periodic motions such as walking and waving hands. While the state sequences were obtained from training samples in this study, estimating state sequences by an EM algorithm is possible [15]. However, it needs more computation cost than the method adopted in our study.

Arguments have swirled around a number of key issues in imitation learning [1], [16]. Among them, *what to imitate* and *how to imitate* are fundamental questions. The first issue is to determine what perceptual aspects are relevant to a task when attempting to imitate a human [16]. The proposed method focused on the estimation of relevant objects in object-manipulation tasks based on the idea that the references of motions performed by humans were implicit. On the other hand, the correspondence problem [17], 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 main focus is the estimation of relevant objects.

B. Related Work

Synthesizing human-like animation and programming humanoid robots to perform new tasks are attractive applications of methods based on the learning-from-observation framework [16]. Inamura *et al.* used HMMs to model motion patterns such as “swing” and “walking” [18]. The trained HMMs were mapped in a space by applying multidimensional scaling (MDS) to Kullback-Leibler divergences between the parameters of the HMMs. Novel motions were generated by the interpolation and the extrapolation of the HMMs in the mapped space [19]. A hidden semi-Markov model (HSMM), which is an HMM with state duration probability distributions, was used for learning walking patterns in [20]. In [21], a Gaussian Process Dynamical Model (GPDM) was proposed and applied to human motion capture data. A GPDM was comprised 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 is able to estimate the optimal intrinsic coordinate system to generalize motion trajectories.

VIII. Conclusion

This paper presented an imitation learning method for object-manipulating motions. Conventional methods for learning object manipulation from demonstrations 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 this paper is providing an algorithm for estimating reference points to overcome such a limitation.

The proposed method was applied to object-manipulation and evaluated its generation errors. We also obtained an accuracy of 90% for motion recognition, which was not presented due to space limitations. The motion recognition results are explained in detail in [9]. Even though 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 be explored, since that is necessary to develop a robot which imitates the goal of an action. Demo video clips are available at `http://mastarpj.nict.go.jp/~ksugiura/video_gallery/video_gallery_en.html`.

Appendix

This appendix explains the trajectory-generation method proposed in [15]. For simplicity, the notation of the world coordinate system shown by the left superscript is omitted in the below, so for example ${}^{W}\lambda$ is written as λ.

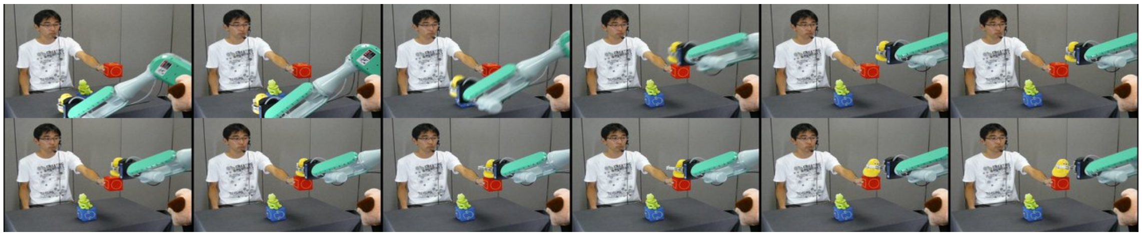

Fig. 10. Sequential photographs of manipulator executing "place-on" manipulation trajectories.

The OPDFs of the HMM λ are Gaussian distributions, therefore Equation (10) can be rewritten as

$$\hat{\Xi} = \underset{\Xi}{\operatorname{argmax}} \log P(\Xi|\boldsymbol{q},\lambda) \tag{17}$$

$$= \underset{\Xi}{\operatorname{argmax}} \left\{ -\frac{1}{2}\Xi^\top \Sigma^{-1}\Xi + \Xi^\top \Sigma^{-1}\boldsymbol{\mu} + K \right\}, \tag{18}$$

where

$$\Sigma^{-1} = \operatorname{diag}\left[\Sigma^{-1}_{q_{L'+1}}, \Sigma^{-1}_{q_{L'+2}}, ..., \Sigma^{-1}_{q_{T-L'}}\right] \tag{19}$$

$$\boldsymbol{\mu} = \left[\boldsymbol{\mu}^\top_{q_{L'+1}}, \boldsymbol{\mu}^\top_{q_{L'+2}}, ..., \boldsymbol{\mu}^\top_{q_{T-L'}}\right]^\top \tag{20}$$

$$L' = \max_k L^{(k)} \tag{21}$$

$\boldsymbol{\mu}_{q_i}$ and Σ_{q_i} denote the $3m \times 1$ mean vector and the $3m \times 3m$ covariance matrix, respectively, and K is a constant. $\boldsymbol{q}$ denotes an arbitrary state sequence, which is set as $\boldsymbol{q} = \bar{\boldsymbol{q}}$ in this study.

Equation (4) can be rewritten in matrix form:

$$\Xi = W\boldsymbol{x}, \tag{22}$$

where

$$\boldsymbol{x} = \left[\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, ..., \boldsymbol{x}_T^\top,\right]^\top$$

$$W = \left[\boldsymbol{w}_{L'+1}, \boldsymbol{w}_{L'+2}, ..., \boldsymbol{w}_{T-L'}\right]^\top$$

$$\boldsymbol{w}_t = \left[\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}\right]$$

$$\boldsymbol{w}_t^{(k)} = \Big[\boldsymbol{0}_{m\times(t-L^{(k)}-1)m}, w^{(k)}(-L^{(k)})I_{m\times m}, ..., w^{(k)}(0)I_{m\times m}, ..., w^{(k)}(L^{(k)})I_{m\times m}, \boldsymbol{0}_{m\times(T-t-L^{(k)})m}\Big]^\top$$

To solve Equation (17) under condition (22), we solve the following equation:

$$\frac{\partial \log P(W\boldsymbol{x}|\boldsymbol{q},\lambda)}{\partial \boldsymbol{x}} = 0 \tag{23}$$

From Equations (18), (22) and (23) we obtain

$$W^\top \Sigma^{-1} W\boldsymbol{x} = W^\top \Sigma^{-1}\boldsymbol{\mu} \tag{24}$$

Equation (24) is solved with $O(Tm^3)$ operations by using Cholesky decomposition.

References

[1] V. Krüger, D. Kragic, A. Ude, and C. Geib, "The meaning of action: a review on action recognition and mapping," *Advanced Robotics*, vol. 21, no. 13, pp. 1473–1501, 2007.

[2] T. Regier, *The Human Semantic Potential: Spatial Language and Constrained Connectionism*. Bradford Books, 1996.

[3] K. Ogawara, J. Takamatsu, H. Kimura, and K. Ikeuchi, "Generation of a task model by integrating multiple observations of human demonstrations," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, 2002, pp. 1545–1550.

[4] ——, "Modeling manipulation interactions by hidden Markov models," in *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, pp. 1096–1101.

[5] A. Billard, S. Calinon, and F. Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 370–384, 2006.

[6] S. Calinon and A. Billard, "A probabilistic programming by demonstration framework handling constraints in joint space and task space," in *Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 367–372.

[7] K. Sugiura and N. Iwahashi, "Learning object-manipulation verbs for human-robot communication," in *Proceedings of the 2007 workshop on Multimodal interfaces in semantic interaction*, 2007, pp. 32–38.

[8] ——, "Motion recognition and generation by combining reference-point-dependent probabilistic models," in *Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 852–857.

[9] K. Sugiura, N. Iwahashi, H. Kashioka, and S. Nakamura, "Learning, generation, and recognition of motions by reference-point-dependent probabilistic models," *Advanced Robotics*, vol. 25, no. 6-7, pp. 825–848, 2011.

[10] "Robocup@home rules & regulations." [Online]. Available: http://www.ai.rug.nl/robocupathome/

[11] R. W. Langacker, *Foundations of Cognitive Grammar: Theoretical Prerequisites*. Stanford Univ Pr, 1987.

[12] L. A. Carlson-Radvansky and D. E. Irwin, "Frames of reference in vision and language: where is above?" *Cognition*, vol. 46, no. 3, pp. 223–244, 1993.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[14] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 349–371, 2003.

[15] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proceedings of ICASSP*, 2000, pp. 1315–1318.

[16] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends in Cognitive Science*, vol. 6, pp. 481–487, 2002.

[17] C. Nehaniv and K. Dautenhahn, "The correspondence problem," in *Imitation in Animals and Artifacts*. MIT Press, 2002, pp. 41–61.

[18] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura, "Embodied symbol emergence based on mimesis theory," *International Journal of Robotics Research*, vol. 23, no. 4, pp. 363–377, 2004.

[19] T. Inamura and T. Shibata, "Geometric proto-symbol manipulation towards language-based motion pattern synthesis and recognition," in *Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 334–339.

[20] N. Niwase, J. Yamagishi, and T. Kobayashi, "Human walking motion synthesis with desired pace and stride length based on HSMM," *IEICE transactions on information and systems*, vol. 88, no. 11, pp. 2492–2499, 2005.

[21] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models," in *Advances in Neural Information Processing Systems*, vol. 18. MIT Press, 2006, pp. 1441–1448.