2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Learning of Motor Skills based on Grossness and Fineness of Movements in Daily-life Tasks

Sang Hyoung Lee[1], Nam Jun Cho[2], and Il Hong Suh[3,†]

Abstract— **In this paper, we propose a novel method for learning motor skills based on grossness and fineness of movements involved in daily-life tasks. Grossness and fineness depend on the degrees of complexity (i.e., linear combinations between basis vectors) and repeatability (i.e., repeat accuracies between multiple trials) of such movements. In such a daily-life task, a robot's movements are usually related to a task-relevant object. Therefore, the complexity and the repeatability should be acquired from datasets that include the spatial and temporal relationships between a robot and a task-relevant object. To measure the degree of complexity, correlations are first obtained from each data by canonical correlation analysis. To measure the degree of repeatability, variations are then obtained from covariances between datasets acquired by multiple trials. The grossness and fineness are finally acquired by combining the correlations and the variations. To learn a motor skill, a Gaussian Mixture Model (GMM) is estimated using well-known methods as Principal Component Analysis (PCA), k-means, Bayesian Information Criterion (BIC), and Expectation-Maximization (EM) algorithms. First, initial parameters of a GMM are estimated by weighting a conventional k-means algorithm with the grossness and fineness. Based on PCA, BIC, and EM algorithms, the GMM is then estimated using the initial parameters and a robot's motion trajectories.**

To validate our proposed methods, the GMM is evaluated in terms of reproduction and recognition using a robot arm that performs two daily-life tasks: cookie-decorating and constrained-delivering tasks.

I. INTRODUCTION

Even after several decades, it remains a challenge for robots to learn a daily-life task although various approaches have been proposed to learn such tasks [1]. In these approaches, several motor skills have been represented as Gaussian Mixture Models (GMMs) using well-known methods such as Principal Component Analysis (PCA), k-means, Bayesian Information Criterion (BIC), and Expectation-Maximization (EM) algorithms. However, it is difficult for a robot to learn such a daily-life task, since these methods focused on not the combinations of fine and gross movements but totally fine (or totally gross) movements [2].

Let us consider an example to intuitively understand the gross and fine movements embedded in a daily-life task. In this example, a robot decorates a tiny cookie with chocolate for providing a human with a chocolate-coated cookie. The task can be roughly divided as follows: (1) the movements that a robot approaches the tiny cookie for decoration; (2) the movements that a robot decorates the tiny cookie with chocolate; and (3) the movements that a robot departs from the tiny cookie after finishing the task. In these movements, the movements associated with (1) and (3) can be identified as gross movements, in comparison with (2). The movements associated with (2) can be considered as fine movements, in comparison with (1) and (3). Likewise, gross and fine movements should be relatively determined according to the task. In this paper, gross and fine movements are defined as follows; i) gross movement: this movement involves simple patterns, though the movement may be varied over large space during a short time interval. It allows flexible reproductions while repeating several trials and ii) fine movement: this movement involves complex patterns (i.e., combinations of simple patterns), though the movement may be varied over smaller spaces during longer time intervals than gross movements. It also allows precise reproductions while repeating several trials.

To learn such tasks better, a robot should therefore be able to represent all its movements in accordance with grossness and fineness, since a daily-life task usually includes gross and fine movements. In fact, it is challenging to represent such gross and fine movements using a fixed criterion. It is possible to consider grossness and fineness by individually modeling segmented movements using segmentation approaches [3]–[5]. However, even such segmentation approaches do not guarantee modeling gross and fine movements, since grossness and fineness are not explicitly considered in these segmentation approaches. To resolve this problem, we proposed a learning method that segments movements based on temporal and spatial entropies in [2]. Even though fine and gross movements were well modeled in [2], there are some drawbacks as follows; in such a daily-life task, a robot's movements are usually related to a task-relevant object. Therefore, the grossness and fineness should be measured from the datasets that include spatial and temporal relationships between a robot and a task-relevant object. However, the previous method did not consider such relationships. In [2], the gross and fine movements are moreover modeled after being iteratively detected and segmented without considering the complexity of movements. Thus, the previous method may regard simple movements (e.g., drawing straight line) as fine movements if the movements are lazily executed, or otherwise. Moreover, it

*This work was supported by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government (MEST)(NRF-M1AXA003-2011-0028553).

[1]S. H. Lee is with the Education Center for Network-based Intelligent Robotics, Hanyang University, Seoul, Korea zelog@hanyang.ac.kr

[2]N. J. Cho is with the Department of Electronics and Computer Engineering, Hanyang University, Seoul, Korea namjun@hanyang.ac.kr

[3,†]I. H. Suh with the Department of Electronics and Computer Engineering, Hanyang University, Seoul, Korea ihsuh@hanyang.ac.kr, All correspondence should be addressed to I. H. Suh.

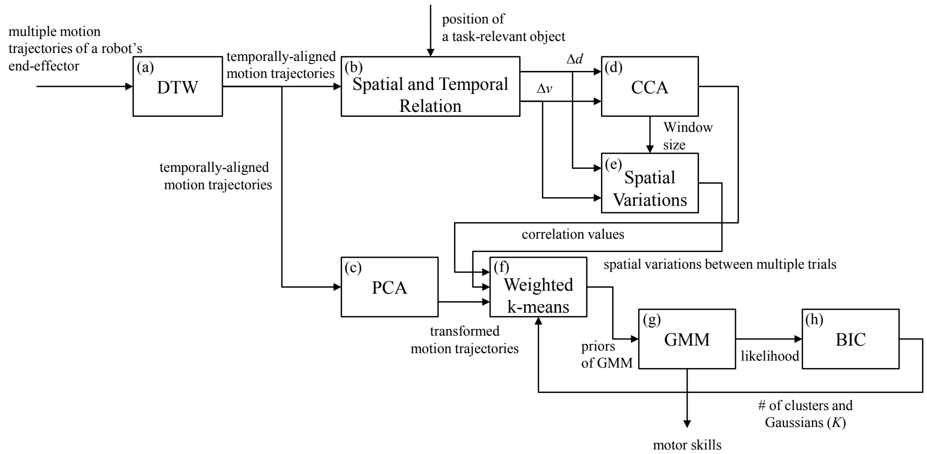

Fig. 1. Eight processes for learning a motor skill based on grossness and fineness; (a) Dynamic Time Warping (DTW) process, (b) process of calculating spatial and temporal relationships between a robot and a task-relevant object, (c) Principal Component Analysis (PCA) process, (d) Canonical Correlation Analysis (CCA) process for calculating correlations of movements, (e) process of calculating variations between multiple demonstrations, (f) process of weighting the k-means algorithm using grossness and fineness, (g) process of learning a Gaussian Mixture Model (GMM) using initial parameters obtained by the weighted k-means algorithm, and (h) Bayesian Information Criterion (BIC) process for determining the numbers of clusters and Gaussians.

is important to determine the threshold values for segmenting and remodeling movements, since the method depends on threshold values of temporal and spatial entropies. To overcome these drawbacks, we propose a novel method for learning a motor skill based on grossness and fineness acquired from the complexity and the repeatability of movements. To use spatial and temporal relationships between a robot and a task-relevant object, especially, the grossness and fineness of movements are measured using following dataset; i) change rate of relative distance between a robot and a task-relevant object and ii) change rate of relative velocity between a robot and a task-relevant object. These are important information to analyze the spatial and temporal relationships between a robot and a task-relevant object.

In this paper, the grossness and fineness depend on the degrees of complexity and repeatability of movements, simultaneously. The rationale is as follows; if a movement may be complex, it should be regarded as a fine movement even though variation is large between multiple trials (i.e., the movement is allowed to be flexibly reproduced during multiple trials). On the other hand, a movement should be considered as a gross movement if it is simple (e.g., drawing straight line) even if its variation is small between multiple trials. The complexity can be measured from correlations of Canonical Correlation Analysis (CCA) by finding linear combinations between basis vectors that maximize their correlations with datasets. In CCA, correlation tends to be low when there are complex patterns in the movements, or otherwise [6]. The repeatability is also measured by variations (i.e., z-scores of the sum of eigenvalues) obtained from covariances between datasets acquired by multiple trials. Finally, the grossness and fineness are acquired by combining the correlations and variations.

The grossness and fineness should be applied to a motor skill. K-means algorithm has been used to acquire initial parameters of a GMM [7]. Here, the conventional k-means algorithm usually focuses on gross movements by a cost function (e.g., Euclidean distance). To consider the characteristics of dataset, some researchers have proposed various cost functions for such a k-means algorithm [8], [9], However, the grossness and fineness have not been considered although these information are important to achieve a daily-life task. Before learning a motor skill, the k-mean algorithm is first weighted by the grossness and fineness. Based on EM, a motor skill is then modeled as a GMM using the initial parameters acquired from the weighted k-means algorithm. In this paper, there are three important contributions to be presented:

i) How to acquire grossness and fineness of movements;
ii) How to use grossness and fineness for learning a motor skill;
iii) Evaluating the performances of our proposed method using some robot experiments.

The rest of this paper is organized as follows: Section II presents the details of learning a motor skill based on grossness and fineness. Section III presents the experimental results of a robot arm performing two daily-life tasks. In this section, we also evaluate our proposed method in terms of reproduction and recognition. Section IV discusses the proposed method. Finally, in section V we present our conclusion and plans for future researches.

II. Learning of Motor Skill Based on Grossness and Fineness

As noted in Section I, gross and fine movements involved in a whole motion trajectory should be modeled for learning a daily-life task. To this end, such grossness and fineness first need to be calculated by combining correlations (i.e., degree of complexity) and variations (i.e., degree of repeatability), after which a motor skill is modeled as a GMM based on grossness and fineness. Fig. 1 shows the entire process involved in making a skilligent robot learns a motor skill based on grossness and fineness.

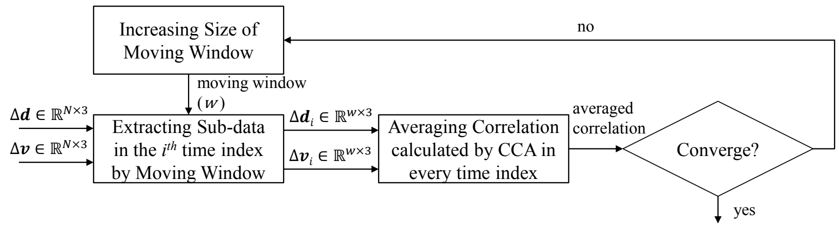

Fig. 2. Processes of determining the size of moving window and calculating correlations by CCA when using the moving window.

A. Preprocessing Processes

In preprocessing processes, some procedures should be first processed as shown in Fig. 1(a)-(c). First, motion trajectories are extracted from a robot's end-effectors through multiple demonstrations. The motion trajectories are temporally aligned by Dynamic Time Warping (DTW) as shown in Fig. 1(a). Next, spatial and temporal relationships between a robot and a task-relevant object are acquired from the positional information of the end-effector and the object, as shown in Fig. 1(b). Here, the relationships indicate the change rates of relative distance and relative velocity between the robot and the object. Finally, PCA is applied for reducing the dimensions of motion trajectories, as shown in Fig. 1(c). Even though PCA is well-known approach for dimension reduction, it is difficult to transform fine movements because of focusing on gross movements. In this paper, PCA is applied for well observing a certain contribution of our proposed learning approach.

B. Processes of Calculating Grossness and Fineness

To calculate grossness and fineness of movements, the complexity of movements is first calculated in each time index, as shown in Fig. 1(d). CCA is an elegant method that measures linear relations between two datasets by finding basis vectors that maximizes their correlations. The complexity of movements can also be measured by CCA. CCA tends to have low correlations when including a lot of linear combinations [6]. In CCA, the correlation is calculated as

$$\rho_i = \frac{\alpha_i^{\mathsf{T}} \Sigma_{i,\Delta d\Delta v} \beta_i}{\sqrt{\alpha_i^{\mathsf{T}} \Sigma_{i,\Delta d\Delta d} \alpha_i} \sqrt{\beta_i^{\mathsf{T}} \Sigma_{i,\Delta v\Delta v} \beta_i}}, \tag{1}$$

where α_i and β_i are sets of basis vectors to calculate the correlation of two datasets in the i^{th} time index, respectively. $\Sigma_{i,\Delta d\Delta v}$, $\Sigma_{i,\Delta d\Delta d}$, and $\Sigma_{i,\Delta v\Delta v}$ indicate the elements associated in the covariance matrix obtained from two datasets in the i^{th} time index. $\Delta d \in \mathbb{R}^{3\times N}$ (i.e., change rate of relative distance) and $\Delta v \in \mathbb{R}^{3\times N}$ (i.e., change rate of relative velocity) are used as two datasets to calculate the complexity of movements. Here, three-dimension indicates Cartesian coordinate, and N is the length of motion trajectories. To measure the correlation, two sets of basis vectors should be obtained by an optimization method. Based on the Lagrange multiplier, the sets of basis vectors, α_i and β_i are calculated as

$$\Sigma_{\Delta d\Delta d}^{-1} \Sigma_{\Delta d\Delta v} \Sigma_{\Delta v\Delta v}^{-1} \Sigma_{\Delta v\Delta d} \alpha_i = \lambda^2 \alpha_i, \tag{2}$$

$$\Sigma_{\Delta v\Delta v}^{-1} \Sigma_{\Delta v\Delta d} \Sigma_{\Delta d\Delta d}^{-1} \Sigma_{\Delta d\Delta v} \beta_i = \lambda^2 \beta_i, \tag{3}$$

where α_i, β_i, and λ are acquired by eigendecomposing $\Sigma_{\Delta d\Delta d}^{-1} \Sigma_{\Delta d\Delta v} \Sigma_{\Delta v\Delta v}^{-1} \Sigma_{\Delta v\Delta d}$ and $\Sigma_{\Delta v\Delta v}^{-1} \Sigma_{\Delta v\Delta d} \Sigma_{\Delta d\Delta d}^{-1} \Sigma_{\Delta d\Delta v}$ of two datasets. The correlation of movements is finally measured based on (1). Likewise, covariances are then acquired to calculate the correlations of movements in each time index using a moving window. To this end, it is important to determine the size of moving window for calculating the correlation in a single time index. Fig. 2 shows the procedures for determining the size of moving window. Using (1), the correlations are iteratively calculated in all time indices while increasing the size of moving window. The size of moving window is determined when conversing the correlation averaged in all time indices, as shown in Fig. 2.

To measure the repeatability of movements, variations are also acquired from covariances between multiple demonstrations in each time index, as shown in Fig. 1(e). To calculate covariance in a single time index, the data is chosen as the size of moving window determined by CCA. The covariance is calculated as

$$\mathbf{C}_i = \mathrm{E}[(\Delta d_i - \Delta \bar{d_i})(\Delta v_i - \Delta \bar{v}_i)], \tag{4}$$

where $\Delta \bar{d_i}$ and $\Delta \bar{v}_i$ are the means of the set of change rates of relative distance and relative velocity in the i^{th} time index, and $\mathrm{E}[\cdot]$ denotes expectation of data. Next, the sum of eigenvalues is obtained from the covariance matrix, $\mathbf{C}_i$. In the covariance matrix, the sum of eigenvalues is obtained as

$$\chi_i = tr(\mathbf{C}_i), \tag{5}$$

where $tr(\cdot)$ is a trace operator for calculating the sum of eigenvalues. The sum of eigenvalues indicates how much variation can be accounted for the multiple demonstrations [10]. Here, the sum of eigenvalues needs to be normalized for being combined with the correlations. Here, the values are normalized by z-scores as

$$\tilde{\delta}_i = \frac{\chi_i - \bar{\chi}_i}{\sigma_i}, \tag{6}$$

where χ_i, $\bar{\chi}_i$, and σ_i are the sum of eigenvalues, its mean, and its standard deviation in the i^{th} time index, respectively. The normalization by z-scores is a well-known method for standardization of values it to have zero mean and unit variance [11]. Finally, the variations, δ_i are acquired by translating the z-scores (to be positive values) for being used as weights.

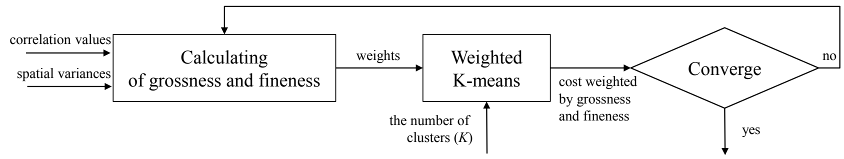

Fig. 3. Processes of the proposed weighted k-means algorithm based on grossness and fineness for acquiring the initial parameters of a GMM.

These are two important criteria for acquiring grossness and fineness, and these criteria should be considered for calculating the grossness and fineness, as noted in Section I. Thus, the grossness and fineness of movements is defined as

$$\omega_i = \frac{1}{\rho_i} \cdot \frac{1}{\delta_i}, \tag{7}$$

where ρ_i and δ_i are the correlations and variations (i.e., the translated z-scores) in the i^{th} time index. Here, the weight tends to be high when the correlation is low (i.e., ρ_i is smaller, and the complexity is high) and the variation is low (i.e., δ_i is smaller, and the repeatability is higher).

C. Processes of Learning Motor Skill Based on Grossness and Fineness

To represent a motor skill, a GMM is modeled using motion trajectories, $\Psi \in \mathbb{R}^{(D'+1)\times N}$, transformed by PCA in preprocessing step. It is important to initialize a GMM using reasonable priors when using EM, since EM does not guarantee global maxima. Using a k-means algorithm is one of renowned approaches to resolve this problem. Before modeling a GMM, in this paper, the grossness and fineness are used for weighting a cost function of a k-means algorithm. Here, such a k-means algorithm tries to acquire the parameters that minimize the sum of variances while the variance in each cluster is calculated using a cost function such as Euclidean distance. Our grossness and fineness are used for regulating the costs as shown in Fig. 1(f). This equation is defined as

$$\varepsilon = \sum_{i=1}^{K} \sum_{j\in S_i} \omega_j \cdot \left|\left|\Psi_j - \mu_i\right|\right|^2, \tag{8}$$

where μ_i and S_i are the center of the i^{th} cluster and the j^{th} data associated in the i^{th} cluster, respectively. $||\cdot||$ is a norm operator for calculating Euclidean distance, and ω_j is the degree of grossness and fineness in the j^{th} data. Here, the number of clusters (K) is finally determined by BIC as shown in Fig. 1(h). By using (7), even though Euclidean distance between the data and its center is close, the cost tends to be increased when ω_j is is higher (i.e., the data is associated in fine movements). As a result, there are more clusters in the fine movements by the weighted k-means algorithm. The initial parameters are acquired for initializing a GMM. Fig. 3 shows the iteration process of our weighted k-means algorithm.

A GMM is modeled using the EM algorithm while using initial parameters provided by the weighted k-means algorithm as shown in Fig. 1(g). Here, the number of Gaussians is also determined by BIC, as shown in Fig. 1(h). In expectation step of EM, the likelihood is calculated as

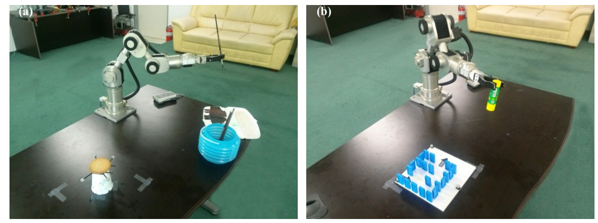

Fig. 4. Illustrations of the cookie-decorating and the constrained-delivering tasks; (a) the experimental setting of cookie-decorating task and (b) the experimental setting of constrained-delivering task.

$$P(\Psi,\mathbf{z}|\theta) = \prod_{i=1}^{N} \sum_{j=1}^{K} \mathbb{I}(z_i = j)\tau_j f(\Psi_i;\mu_j,\Sigma_j), \tag{9}$$

where θ includes the K number of parameters, and τ_j, μ_j, and Σ_j indicate the weight, the mean, and the covariance of the j^{th} Gaussian, respectively. Here, $\mathbb{I}$ and f are an indicator function and a probability density function. In EM process, the indicator function $\mathbb{I}$ is usually used as values that have one or zero [12]. Here, the GMM can be modeled using the indicator function regulated by the grossness and fineness without using the weighted k-means algorithm. In our experience, however, the grossness and fineness are more reasonable to apply to the k-means algorithm than the EM algorithm.

III. Experimental Results and Evaluations

To validate our proposed learning method, two daily-life tasks were performed using a robot arm; cookie-decorating and constrained-delivering tasks. In the cookie-decorating task, a robot performs the task using a nominal sequence as follows: first, the robot approaches the tiny cookie. Next, the robot decorates the cookie by painting a pentagram using a brush stained with chocolate. Finally, the brush is withdrawn from the cookie after finishing the decorations. The constrained-delivering task is also performed as follows: first, the robot approaches a labyrinth-plate to pass through the plate. Next, a hand-held object is delivered while ensuring that the object does not brush against the walls of the labyrinth. Finally, the carried object is withdrawn from the labyrinth.

To extract training data and test data, three-dimensional motion trajectory (i.e., (x, y, z) of an end-effector) are first recorded at 25Hz from a robot arm called as Katana (developed by Neuronics). Moreover, positions of task-relevant objects (i.e., (x, y, z) of a cookie and a labyrinth-plate) are

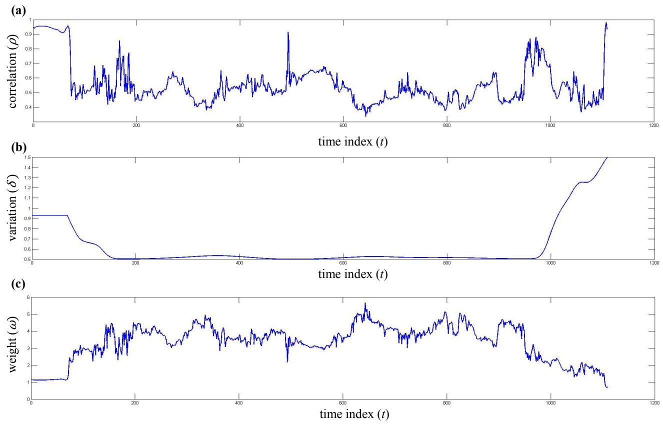

Fig. 5. Correlations, variations, and weights (i.e., grossness and fineness) obtained in the cookie-decorating task; (a) correlations obtained by CCA, (b) variations by the translated z-scores of eigenvalues in covariances acquired from multiple demonstrations, and (c) weights by combining the inverses of correlations and inversed variations.

extracted using twelve V100:R2 motion capture cameras developed by Optitrack. Fig. 4 illustrates experimental settings of the two tasks. In each task, six demonstrations are acquired without changing the positions of the objects using a kinesthetic teaching method. In each of the six demonstrations, three motion trajectories are used for learning motor skills and the rest are used for evaluating the learned motor skills.

A. Experimental Results

Figures 5–8 show the data perceived from the procedures for learning a motor skill, based on the grossness and fineness of movements embedded in the cookie-decorating task. The motion trajectories of an end-effector extracted from three demonstrations are temporally aligned by DTW. The change rates of relative distance and relative velocity are calculated using the information of the robot and the task-relevant object, after which the correlations of movements are obtained by CCA in each time index as shown in Fig. 5(a). Further, variations between multiple demonstrations are acquired by z-scores of the eigenvalues obtained from covariances in each time index, as shown in Fig. 5(b). Here, the size of moving window is automatically chosen as those at which the averaged correlations are converged while the sizes are incrementally increased. The sizes are 73, 78, and 79 in each of the demonstrations in the cookie-decorating task, respectively. To acquire grossness and fineness of movements, the weights are finally calculated by combining the correlations with the variations, as shown in Fig. 5(c). The movements in which the weights are higher are considered as fine movements, or otherwise. Before learning a GMM, the temporally aligned motion trajectories, $\Psi \in \mathbb{R}^{(3+1)\times 1100}$ are transformed using PCA. Here, four dimensions indicate a three-dimensional spatial variable in a latent space and a one-dimensional temporal variable, and 1100 is the length of the motion trajectories. In fact, the dimensionality of motion trajectories in the latent space is not changed in this experiment, since we use eigenvectors in which the sum of

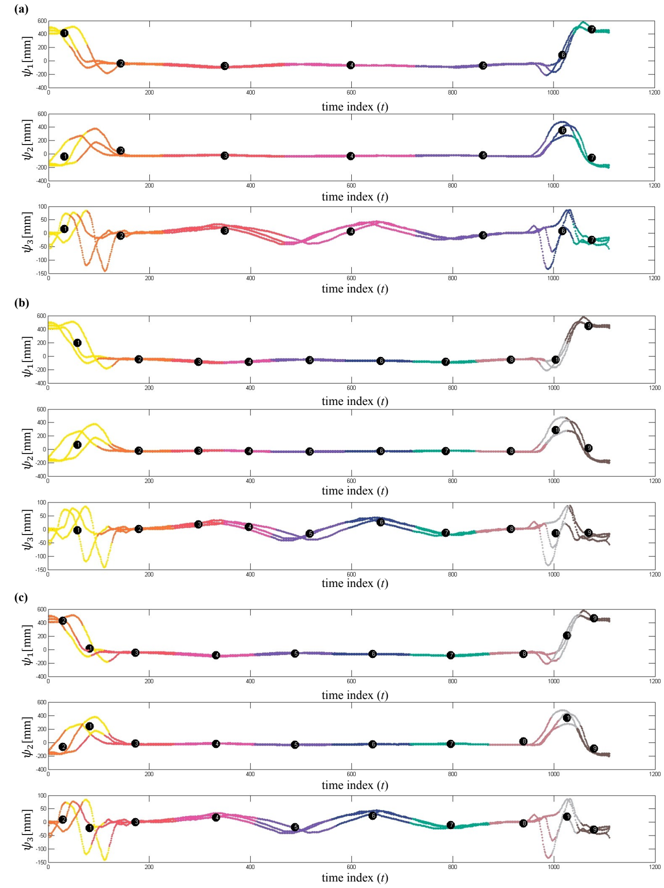

Fig. 6. Results of k-means algorithms of three types in the cookie-decorating task; (a) the conventional k-means and BIC algorithms, (b) our weighted k-means and BIC algorithms, and (c) the overfitted k-means algorithm. Here, the colors indicate different clusters, and the dotted lines indicate motion trajectories of training data.

eigenvalues is 0.98.

Next, initial parameters of a GMM are estimated by the k-means algorithms of three types; (1) the conventional k-means and BIC algorithms, (2) our weighted k-means and BIC algorithms, and (3) overfitted k-means algorithm (i.e., the same number of clusters with (2) without using BIC). Fig. 6 shows the results of k-means algorithms of three types. In the case of (2), the number of clusters is larger than in the case of (1), as shown in Fig. 6(b). Here, the extra clusters of (2) are estimated to assign the fine movements (i.e., decorating) than the case of (1), since the costs are increased by our proposed method when the weights are higher although Euclidean distance is small. In fact, the clusters should be assigned in such fine movements, since the fine movements are important for achieving the task. In the case of (3), the extra clusters are usually assigned in gross (i.e., approaching and departing) movements when comparing with the case of (1), as shown in Fig. 6(c), since the cost tends to high in the gross movements by the conventional k-means algorithm. Based on EM, the GMM is modeled using the initial parameters obtained by k-means algorithms of three types as shown in Fig. 7. Fig. 7(a)-

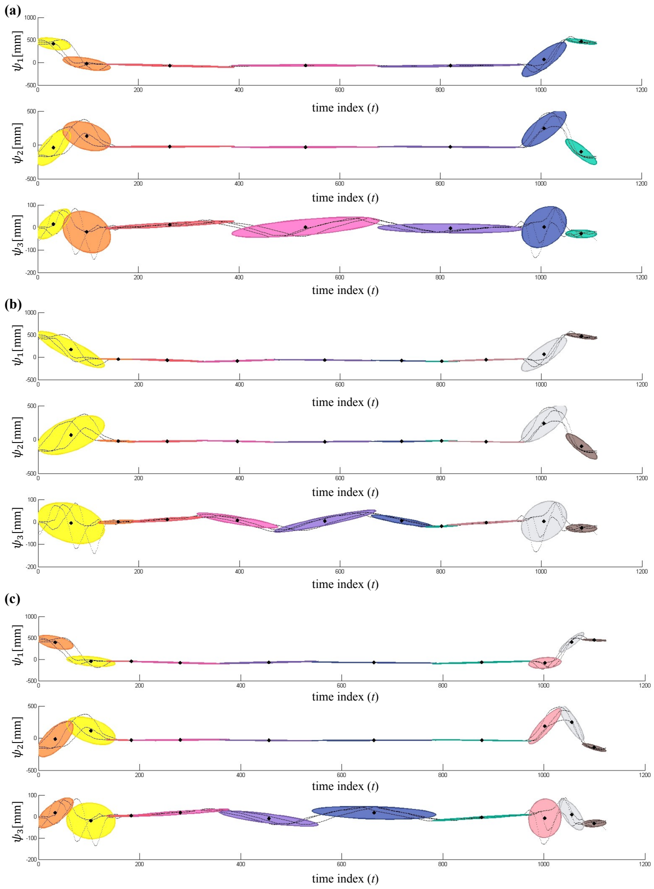

Fig. 7. Learned GMMs on using the parameters initialized by the k-means algorithms of three types in the cookie-decorating task; (a) the GMM initialized by the conventional k-means and BIC algorithms, (b) the GMM initialized by our weighted k-means and BIC algorithms, and (c) the GMM initialized by the conventional k-means algorithm after being overfitted. Here, the numbers of GMMs are seven, ten, and ten, respectively. The ellipsoids are all Gaussians, and the dotted lines indicate motion trajectories of training data.

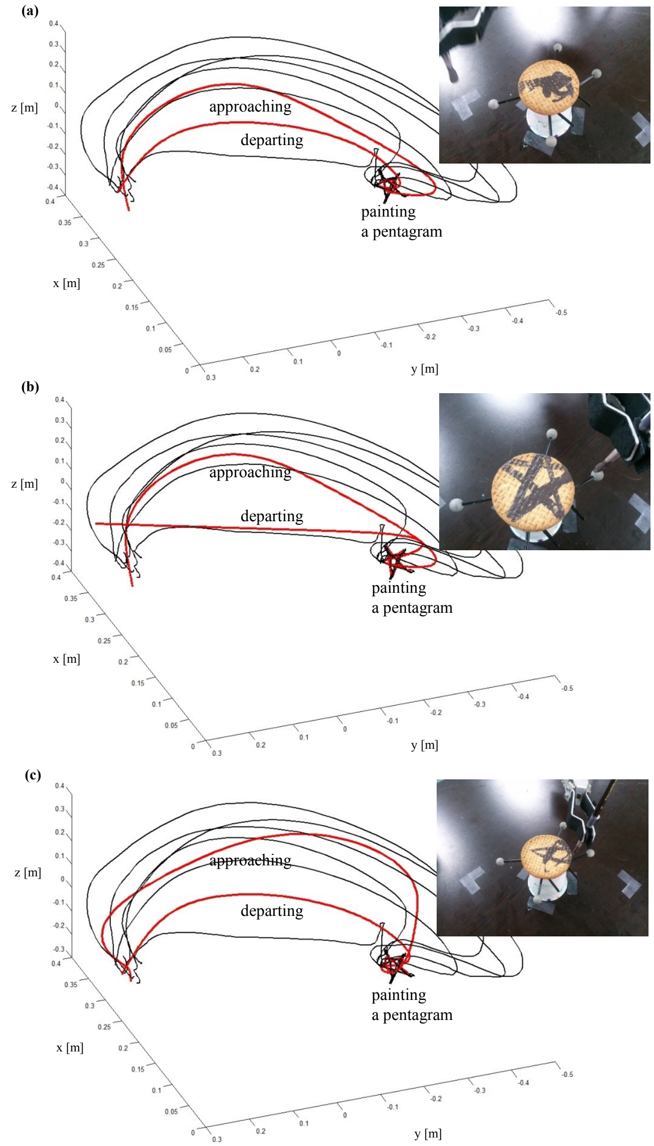

Fig. 8. Motion trajectories retrieved by GMR when using three GMMs and their results executed by a robot arm in the cookie-decorating task; (a) the retrieved motion trajectory and its result when using the GMM of Fig. 7(a), (b) the retrieved motion trajectory and its result when using the GMM of Fig. 7(b), and (c) the retrieved motion trajectory and its result when using the GMM of Fig. 7(c). Here, black lines indicate motion trajectories acquired by three demonstrations, and red lines indicate the motion trajectories retrieved from GMMs.

(c) illustrates three GMMs initialized by the parameters of Fig. 6(a)-(c), respectively. The GMMs are modeled to have similar characteristics according to the k-means algorithms.

B. Evaluations

The GMMs should be able to retrieve as well as recognize motion trajectories. Therefore, these reproduction and recognition abilities are evaluated from the learned GMMs using test data. To evaluate the reproduction ability of three GMMs, the motion trajectories are retrieved by Gaussian Mixture Regression (GMR). Fig. 8 shows the motion trajectories retrieved from three GMMs and their results when using the motion trajectories. The GMMs of Fig. 7(a) and (c) cannot achieve the cookie-decorating task, since the motion trajectories retrieved by GMR cannot paint a pentagram on the cookie, as shown in Fig. 8(a) and (c). On the other hand, the GMM of Fig. 7(b) can retrieve the motion trajectories to paint the pentagram, as shown in Fig. 8(b).

To validate the reproduction ability, additionally, the constrained-delivering task is performed according to the same procedures as the cookie-decorating task. Here, the sizes of moving windows are determined as 69, 88, and 87 in all training data, respectively. Fig. 9 shows the GMMs learned using initial parameters from the k-means algorithms of three types. As in the cookie-decorating task, the GMMs of Fig. 9(a) and (c) cannot achieve the task, since the motion trajectories cannot be reproduced to pass through the labyrinth without crashes, as shown in Fig. 10(a) and (c). On the other hand, the GMM of Fig. 9(b) reproduces the motion trajectories that can be successfully passed in the labyrinth, as shown in Fig. 10(b). As a result, it is possible for a robot to achieve a daily-life task when the robot learns a motor skill after considering the gross and fine movements embedded in

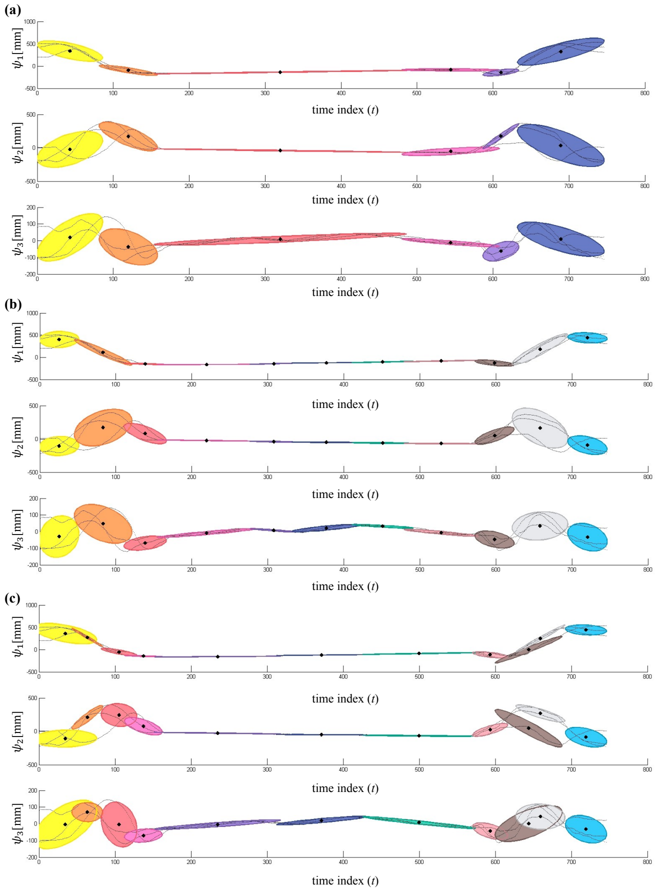

Fig. 9. Learned GMMs when using the parameters initialized by the k-means algorithms of three types in the constrained-delivering task; (a) the GMM initialized by the conventional k-means and BIC algorithms, (b) the GMM initialized by our weighted k-means and BIC algorithms, and (c) the GMM initialized by the conventional k-means algorithm after being overfitted. Here, the numbers of GMMs are six, eleven, and eleven, respectively. The ellipsoids are all Gaussians, and the dotted lines indicate the motion trajectories of training data.

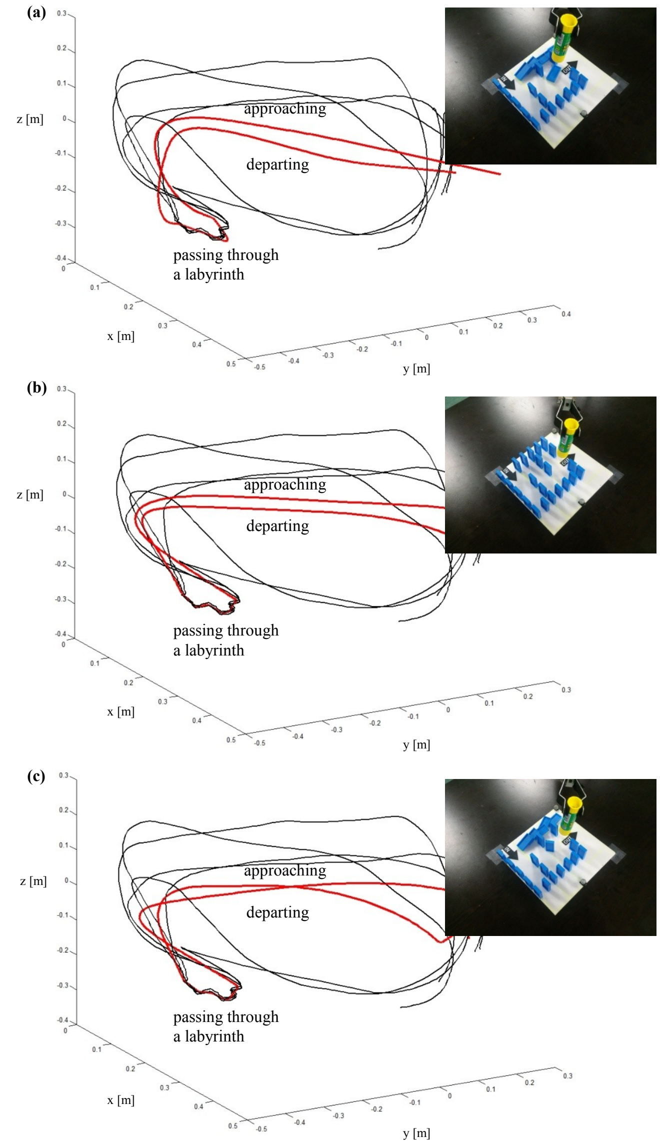

Fig. 10. Motion trajectories retrieved by GMR when using three GMMs and their results executed by a robot arm in the constrained-delivering task; (a) the retrieved motion trajectory and its result when using the GMM of Fig. 9(a), (b) the retrieved motion trajectory and its result when using GMM of Fig. 9(b), and (c) the retrieved motion trajectory and its result when using the GMM of Fig. 9(c). Here, black lines indicate motion trajectories acquired by three demonstrations, and red lines indicate the motion trajectories retrieved from GMMs.

the tasks.

To evaluate the recognition ability, moreover, the log-likelihoods are estimated and compared from the GMMs of three types using test data. The GMMs obtained from our weighted k-means and the overfitted k-means algorithms are obviously overfitted as compared to the GMM obtained from the conventional k-means algorithm, because the GMMs are more number of Gaussians than the GMM estimated by the conventional k-means and BIC algorithms. In the cookie-decorating task, the log-likelihoods are estimated from three GMMs using three test data, as shown in Fig. 11(a). The GMM of Fig. 7(a) can be used as a reference, sine the GMM is estimated using the conventional k-means and BIC algorithms. The log-likelihoods (i.e., red bars) estimated from the GMM of Fig. 7(c) are naturally lower than the results (i.e., blue bars) obtained from the GMM of Fig. 7(a), since the GMM of Fig. 7(c) is overfitted using more parameters. Even though the GMM of Fig. 7(b) is also overfitted using the same number of Gaussians as in the GMM of Fig. 7(c), its performance (i.e., green bars) is better than that of the GMM of Fig. 7(c). This rationale is the GMM of Fig. 7(b) is only overfitted in the fine movements that the spatial variations are very small even in the test data. In some test data, surprisingly, the estimation performances obtained from the GMM of Fig. 7(b) are better than the results obtained from the GMM of Fig. 7(a) although the GMM of Fig. 7(b) is fitted using more parameters. It is because the log-likelihood is very high in the Gaussians in which the fine movements are modeled. In fact, such Gaussians tend to allow very small spatial variations, even in the set of test data. Likewise, the GMMs show the similar performances even in the constrained-delivering task, as shown in Fig. 11(b).

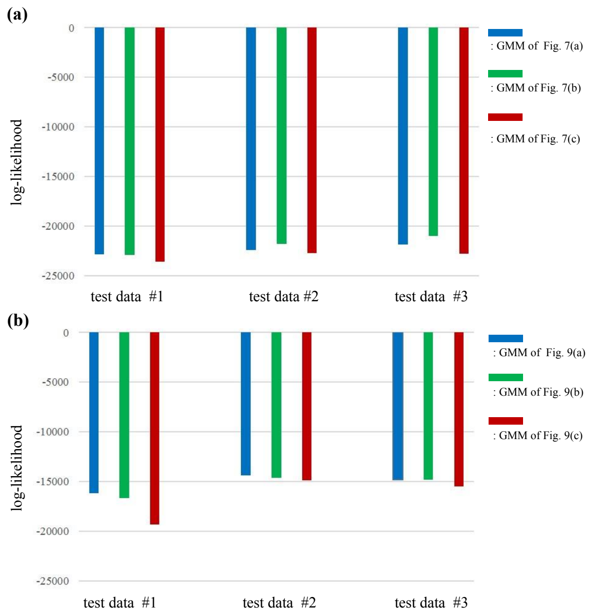

Fig. 11. Log-likelihoods estimated from the GMMs of three types when using the test data in the cookie-decorating and the constrained-delivering tasks; (a) the log-likelihoods estimated from three GMMs in the cookie-decorating task and (b) the log-likelihoods estimated from three GMMs in the constrained-delivering task. Here, blue, green, and red bars indicate the results of GMMs estimated when using the conventional k-means, our weighted k-means, the overfitted k-means algorithms, respectively.

As a result, our proposed learning method, which considers grossness and fineness, makes it possible for task-achievable motion trajectories to be retrieved as compared with the conventional models, whereas the estimation performance is the better than the overfitted models.

IV. Discussion

In our earlier work, we proposed a novel method for modeling such gross and fine movements by segmenting and remodeling the movements, iteratively. However, it is not easy to model such gross and fine movements using our previous method in the cases of complex movements that are quickly executed or simple movements that are slowly executed, since it used temporal and spatial entropies without explicitly considering the complexity and repeatability.

In this paper, correlations and variations were calculated in each time index. For this, the sizes of moving windows should be determined for calculating the values in a single time index. In our experiments, the sizes of moving windows are chosen as those at which the averaged correlations are converged while the sizes are incrementally increased. This rationale is the complexity of movements can be best noticed in the size that the averaged correlation is maximized.

Our proposed method was used for modeling GMMs. However, the core paradigm can also be used for segmenting movements as in [5]. Moreover, the learning method can also be used for applying well-known skill learning methods such as Hidden Markov Model (HMMs) and Dynamic Movement Primitives (DMPs). In the learning process of HMMs, the grossness and fineness can be used to initial parameters for Baum-Welch algorithm. For DMPs, the grossness and fineness can be used for adjusting the arrangements of the Gaussian basis functions, because the forcing terms of DMPs are fitted by the basis functions.

V. Conclusion and Future Works

In this paper, we have proposed a novel method for learning a motor skill, based on grossness and fineness. For this, the correlations of movements are first acquired by CCA, and then the variations are obtained by z-scores of the sum of eigenvalues from the covariances between multiple trials. The grossness and fineness are finally acquired by combining the correlations with the variations. The grossness and fineness are used for weighting the k-means algorithm, after which the GMM is estimated using initial parameters obtained from the weighted k-means algorithm.

The reproduction and recognition abilities are evaluated using two daily-life tasks; the cookie-decorating and the constrained-delivering tasks. The learned GMM can retrieve the reasonable motion trajectories, while maintaining the estimation performance although the GMM is overfitted when comparing the conventional GMM.

In our future work, we intend to apply our scheme to various types of motion trajectories such as force/torque. Furthermore, we aim to apply this learning method to various existing methods such as HMMs and DMPs.

References

[1] S. Calinon, Robot Programming by Demonstration, Springer Handbook of Robotics, Springer, pp. 1371-1394, 2008.
[2] S. H. Lee, G. N. Han, I. H. Suh, B. J. You, "Skill Learning Using Temporal and Spatial Entropies for Accurate Skill Acquisition," in Proc. of IEEE Intl Conf. on Robotics and Automation (ICRA), pp.1315-1322, 2013.
[3] P. Pastor, M. Kalakrishnan, F. Meier, F. Stulp, J. Buchli, E. Theodorou, S. Schaal, "From Dynamic Movement Primitives to Associative Skill Memories," Robotics and Autonomous Systems, pp. 351-361, 2012.
[4] D. Kulic, W. Takano, and Y. Nakamura, "Online Segmentation and Clustering from Continuous Observation of Whole Body Motions," IEEE Transactions on Robotics, vol. 25, no. 5, pp.1158-1166, 2009.
[5] S. H. Lee, I. H. Suh, S. Calinon, R. Johansson, "Learning Basis Skills by Autonomous Segmentation of Humanoid Motion Trajectories," in Proc. of IEEE Intl Conf. on Humanoid Robots, pp.112-119, 2012.
[6] D. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical Correlation Analysis: An Overview with Application to Learning Methods," Neural Computation, vol. 16, no. 12, pp. 2639-2664, 2004.
[7] M. Muhlig, M. Gienger, S. Hellbach, J. Steil, C. Goerick, " Task-level Imitation Learning using Variance-based Movement Optimization," in Proc. of IEEE Intl Conf. on Robotics and Automation (ICRA), pp. 1177-1184, 2009.
[8] L. Jing, M. Ng, J. Huang, "An Entropy Weighting K-means Algorithm for Subspace Clustering of High-dimensional Sparse Data," IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 8, pp. 1026-1041, 2007.
[9] K. Kerdprasop, N. Kerdprasop, and P. Sattayatham, "Weighted K-means for Density-biased Clustering," Data Warehousing and Knowledge Discovery, pp. 488-497. 2005.
[10] D. Borcard, P. Legendre, and P. Drapeau, "Partialling Out the Spatial Component of Ecological Variation," vol. 73, no. 3, pp. 1045-1055, 1992.
[11] J. Yang, S. Wang, N. Chen, X. Chen, and P. Shi, "Wearable Accelerometer based Extendable Activity Recognition System," in Proc. of IEEE Intl Conf. on Robotics and Automation (ICRA), pp. 3641-4647, 2010.
[12] T. Bailey and C. Elkan, "Fitting A Mixture Model by Expectation-Maximization to Discover Motifs in Biopolymers," UCSD Technical Report CS94-351, pp.7-8, 1994.