International Journal of Control, Automation, and Systems (2012) 10(1):126-135
DOI 10.1007/s12555-012-0114-1 http://www.springer.com/12555

Incremental Motion Learning through Kinesthetic Teachings and New Motion Production from Learned Motions by a Humanoid Robot

Sumin Cho and Sungho Jo*

Abstract: This work presents a new incremental motion learning algorithm through kinesthetic teachings and a new motion production algorithm by combining learned motions in a humanoid robot. The proposed algorithms are useful for improving the motions that a humanoid robot can produce. The learning algorithm consists of data encoding, time alignment, dimensional reduction, parameter estimation in the Gaussian mixture model (GMM) of motions, GMM refinement, and motion generation steps. The overall procedure is built to be incremental. No historic data memorization is required in any step, and model parameters are enough information to generate motions. The motion production algorithm allows a robot to extract new motions simply from learned motions without requiring teaching sessions. A series of experiments with a humanoid robot serves to validate the performance of the proposed algorithms.

Keywords: Gaussian mixture model, incremental learning, motion learning, humanoid robot.

1. INTRODUCTION

Robotic learning of motions by imitation or through demonstrations has been a commonly investigated as a strategy for human-robot friendly interaction, especially as a way of teaching motions to a robot without serious programming [1-5]. A robot observes multiple demonstrations of a specific skill or is taught kinesthetically over teaching sessions. Then, the robot extracts a generalized motion out of the demonstrations or teachings. The overall procedure can be described as a motion learning algorithm. Existing literature summarizes these overall issues well [6-9].

This paper focuses on kinesthetic teaching, which is a teaching strategy that involves a teacher moving robot limbs directly to implement a specific motion while a robot encodes the motion [10]. The teaching strategy is convenient because it does not require any external sensors, such as a motion tracker, other than internal encoders in a robot. Regardless of whether extra sensors are used or not, the principle of a learning algorithm remains the same.

Even though learning from demonstrations or through kinesthetic teaching is easy to learn and convenient, encoding every motion through this scheme seems inefficient. Learning some motions may allow other motions to be reproduced without full learning. Imagine that a novice dancer exercises upper and lower body gestures separately to master a dancing gesture or learns an overall dancing performance step by step. This enables the dancer to create elaborate dance performances by adding or deducting specific local motions. Take, as another example, a novice golfer practicing a tee shot by separating the motion into take-back, down-swing, and follow-through. It is not necessary to learn the whole motion pattern fully from demonstrations or teaching processes. Such examples inspire us to investigate motion production from learned motions. The motion production scheme is useful for increasing the number of reproducible motions in a robot quickly and for diversifying the motions easily.

By combining these two strategies (motion learning from kinesthetic teachings and motion production from learned motions), a robot can generate a range of complex motions easily. Such a technique may improve dramatically the performances of robots. This work aims to propose a new approach for incremental motion learning and motion production from learned motions in a robot.

The remainder of this paper is organized as follows. Section 2 illustrates the overall steps of incremental motion learning from kinesthetic teaching. Section 3 explains new motion production from learned motions by fusing different joint trajectories spatially or by connecting two learned motions successively. The proposed approach is evaluated through experiments in Section 4, and this work is concluded in Section 5.

2. INCREMENTAL MOTION LEARNING AND GENERATION

Our learning algorithm is fundamentally based on a framework that combines the Gaussian Mixture Model (GMM) with the Gaussian Mixture Regression (GMR)

Manuscript received February 16, 2011; revised July 19, 2011 and September 15, 2011; accepted November 22, 2011. Recommended by Editorial Board member Youngjin Choi under the direction of Editor Hyoukryeol Choi.

This work was supported by the Korea government (MEST) under a KRF grant (No. 2010-0015226) and by the Korea government (MKE) under the Human Resources Development Program for Convergence Robot Specialists.

Sumin Cho and Sungho Jo are with the Department of Computer Science, KAIST, 291 Daehakro, Daejeon, Korea (e-mails: {zerachiel7, shjo}@kaist.ac.kr).
* Corresponding author.

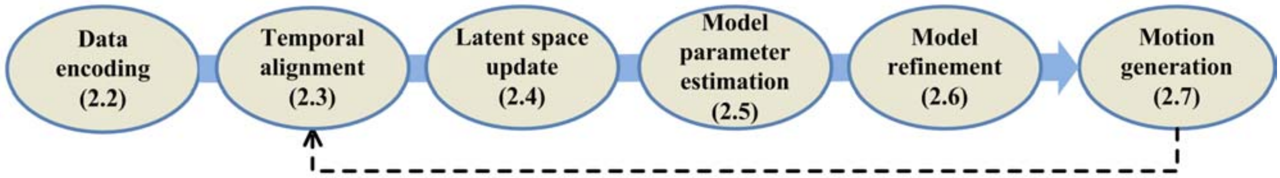

Fig. 1. Incremental motion learning algorithm overview. Each related section number is indicated.

[1,11]. This approach is advantageous because it handles recognition and reproduction steps in a common probabilistic framework, while the two step processes can be distinct. Our approach possesses the advantages directly. In addition, our approach addresses incremental learning of the GMM with no assumptions about the model parameters. The number of Gaussian components in the GMM is not initially fixed but is obtained autonomously during the learning process. The incremental temporal alignment is integrated to align motion data. The Principal Component Analysis (PCA) is applied to build a compact representation by reducing the dimensional space of motion [12,13]. Motion data are projected onto a low-dimensional latent space to maintain a satisfying motion representation. In our approach, the latent space of motion is refined whenever a new kinesthetic teaching is provided. Previous studies [1,11] require specific assumptions, such as a fixed latent space and a predetermined number of GMM parameters, before learning.

2.1. Motion model

The Gaussian Mixture Model (GMM) has popularly been used as a probabilistic model framework to represent a group of similar motion trajectories [1,11]. The probabilistic density function is represented by a combination of K Gaussian components.

$$p(x)=\sum_{k=1}^{K}\pi_k\, p(x\mid k), \tag{1}$$

where π_k is the prior probability, and

$$p(x\mid k)=\mathcal{N}(x;\mu_k,\Sigma_k) =\frac{1}{\sqrt{(2\pi)^d\,|\Sigma_k|}}\exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right),$$

where μ_k and Σ_k are the mean vector and covariance matrix, respectively, of the k-th component.

π_k, μ_k and Σ_k for $k=1,2,\ldots,K$ are model parameters.

Previous works [11,14] proposed that a generalized motion from the GMM can be computed by the Gaussian Mixture Regression (GMR). The temporal and spatial value decomposition of the model parameters are $\mu_k=[\mu_{t,k}\ \mu_{s,k}]^T$ and $\Sigma_k=\begin{bmatrix}\Sigma_{tt,k} & \Sigma_{st,k}\\ \Sigma_{ts,k} & \Sigma_{ss,k}\end{bmatrix}$.

Thus, the GMR gives the following functions [11,14]:

$$\mu_s(t)=\sum_{k=1}^{K}\frac{\pi_k\mathcal{N}(t;\mu_{t,k},\Sigma_{tt,k})}{\sum_{i=1}^{K}\pi_i\mathcal{N}(t;\mu_{t,i},\Sigma_{tt,i})}(\mu_{s,k}+\Sigma_{st,k}\Sigma_{tt,k}^{-1}(t-\mu_{t,k})),$$

$$\Sigma_{ss}(t)=\sum_{k=1}^{K}\left(\frac{\pi_k\mathcal{N}(t;\mu_{t,k},\Sigma_{tt,k})}{\sum_{i=1}^{K}\pi_i\mathcal{N}(t;\mu_{t,i},\Sigma_{tt,i})}\right)^2(\Sigma_{ss,k}-\Sigma_{st,k}\Sigma_{tt,k}^{-1}\Sigma_{ts,k}). \tag{2}$$

New parameters with the time value are respectively defined as:

$$\mu(t)=[t\quad \mu_s(t)]^T,\qquad \Sigma(t)=\begin{bmatrix}0 & 0\\ 0 & \Sigma_{ss}(t)\end{bmatrix}, \tag{3}$$

where $\mu(t)$ represents the regressed trajectories in the latent space.

This work proposes expressing accumulated information after the j-th teaching session compactly as $\{G^j, L^j\}$ where $G^j=\{\pi_k^j,\ \mu_k^j,\ \Sigma_k^j\}_{k=1}^{K}$ is a collection of the GMM parameters in the latent space, and $L^j=\{v^j,P^j\}$ consists of parameters to describe the latent space. Details on L^j are provided in Section 2.4.

Fig. 1 illustrates the overall procedure of our algorithm. The following sections will explain each step.

2.2. Kinesthetic teaching data acquisition

A teacher holds robot limbs and moves them to generate a desired motion in each kinesthetic teaching session. The motion is recorded in terms of 3D joint angle trajectories through encoders on the robot. The collected joint angle trajectory information is represented by $\theta^j=\{\theta^j(i)\}_{i=1}^{N^j}=\{\theta_t^j(i),\ \theta_s^j(i)\}_{i=1}^{N^j}$, where N^j is the total number of data points from j-th kinesthetic teaching. Each $\theta^j(i)$ consists of a time value $\theta_t^j(i)$ and a posture vector $\theta_s^j(i)\in R^d$, where d is the dimensionality of the joint space.

2.3. Incremental temporal alignment

Each sensed teaching trajectory is different in sample length. To apply a learning framework, the motion trajectories are temporally aligned to the same length by applying a dynamic time warping (DTW) algorithm [15-17]. DTW is a template based dynamic programming matching algorithm. It assigns a template trajectory and aligns other trajectories with respect to the template by finding a warping path trajectory that minimizes distance from the template trajectory. In our case, encoded teaching trajectories are not available immediately. Therefore, our template trajectory is not fixed but is refined incrementally as further teaching data becomes available. We propose using the learned motion $\phi^{j-1} = \{\phi^{j-1}(i)\}_{i=1}^{N}$ until the (j-1)-th kinesthetic teaching (see (21)) as a template trajectory for the j-th kinesthetic teaching. Its time component is represented by $\phi_t^{j-1}(i)$ and a posture vector $\phi_s^{j-1}(i)$. For the first teaching session, no template trajectory is required. For the j-th teaching session, using the DTW, a set of matching pairs between the template and the j-th kinesthetic teaching is found. Assuming the number of elements in the set is M, and each element is expressed by $(\varphi(l), \vartheta(l))$, $l = 1,\ldots,M$, where $\varphi(l)$ is from ϕ^{j-1}, and $\vartheta(l)$ is from θ^j, each corresponding time value is estimated as

$$\tau(l) = \frac{j-1}{j}\varphi_t(l) + \frac{1}{j}\vartheta_t(l) \tag{4}$$

where subscript t indicates its time component.

Therefore, each aligned data point is expressed as $[\tau(l)\ \vartheta(l)]^T$. However, its time interval is inconsistent, and the number of data points can be unfixed depending on the matching condition.

According to the DTW method, the matching set should always contain the first and the last data points to satisfy the boundary condition. Therefore,

$$(\varphi(1), \vartheta(1)) = (\phi^{j-1}(1), \theta^j(1)) \text{ and}$$
$$(\varphi(M), \vartheta(M)) = (\phi^{j-1}(N), \theta^j(N^j)).$$

Each motion trajectory begins at time zero, $\tau(1) = 0$, and the last time value is $\tau(M) = \frac{j-1}{j}\varphi_t(M) + \frac{1}{j}\vartheta_t(M)$. If a new time step is set as $t(i) = (i-1)\frac{\tau(M)}{(N-1)}$, $i = 1,\ldots,N$, by considering a constant interval $\frac{\tau(M)}{(N-1)}$, the total sample number is fixed to be N over different teachings. Aligned data points with the fixed interval can be generated from $\{\vartheta(l)\}_{l=1}^{M}$ using a linear interpolation function defined as

$$f(t) = \frac{t-\tau(p)}{\tau(p+1)-\tau(p)}\vartheta_s(p+1) + \frac{\tau(p+1)-t}{\tau(p+1)-\tau(p)}\vartheta_s(p), \tag{5}$$

where $\tau(p)$ is selected among $\tau(l)$, $l = 1,\ldots,M$, to be a time step value that is nearest to but not greater than t, and $\vartheta_s(p)$ is a posture vector in $\vartheta(p)$. The interpolation function describes an aligned spatial posture vector at each time. Let $\xi^j = \{\xi^j(i)\}_{i=1}^{N}$ denote the aligned data points of the j-th kinesthetic teaching:

$$\xi^j(i) = [t(i)\quad f(t(i))]^T. \tag{6}$$

To modify model parameters after the j-th kinesthetic teaching accordingly to the new time alignment, a function is declared to interpolate between the pre-aligned and the aligned time values as follows

$$g(t) = \frac{t-\tau(p)}{\tau(p+1)-\tau(p)}\varphi_t(p+1) + \frac{\tau(p+1)-t}{\tau(p+1)-\tau(p)}\varphi_t(p), \tag{7}$$

where $\varphi_t(p)$ is a time value in $\varphi(p)$. In addition, two regression functions $\mu_s^{j-1}(t)$ and $\Sigma_{ss}^{j-1}(t)$ are declared as described in (2). Similar to (3), $\{\hat{\mu}^{j-1}(i), \hat{\Sigma}^{j-1}(i)\}_{i=1}^{N}$ in the latent space is adjusted taking into account the new time alignment as follows

$$\begin{aligned} \hat{\mu}^{j-1}(i) &= \begin{bmatrix} t(i) & \mu_s^{j-1}(g(t(i))) \end{bmatrix}^T, \\ \hat{\Sigma}^{j-1}(i) &= \begin{bmatrix} 0 & 0 \\ 0 & \Sigma_{ss}^{j-1}(g(t(i))) \end{bmatrix}. \end{aligned} \tag{8}$$

As a result, we obtain the aligned data trajectory $\{\xi^j(i)\}_{i=1}^{N}$ and a parameter set $\{\hat{\mu}^{j-1}(i), \hat{\Sigma}^{j-1}(i)\}_{i=1}^{N}$ for the GMR that represents previous data information inclusively.

2.4. Dimensional reduction

The parameter set described in the latent space is updated to include the most recent data $\{\xi^j(i)\}_{i=1}^{N}$ in the joint space. For the inclusion, let $\{\hat{\mu}^{j-1}(i), \hat{\Sigma}^{j-1}(i)\}_{i=1}^{N}$ be transformed to $\{m^{j-1}(i), C^{j-1}(i)\}_{i=1}^{N}$ in the joint space using the latent space infromation $L^{j-1} = \{v^{j-1}, P^{j-1}\}$ obtained from previous teaching session.

$$\begin{aligned} m^{j-1}(i) &= P^{j-1}\hat{\mu}^{j-1}(i) + v^{j-1}, \\ C^{j-1}(i) &= P^{j-1}\hat{\Sigma}^{j-1}(i)P^{j-1^T}. \end{aligned} \tag{9}$$

Thus, the following equations yield the update of the parameter set:

$$\begin{aligned} u^j(i) &= \frac{j-1}{j}m^{j-1}(i) + \frac{1}{j}\xi^j(i), \\ Q^j(i) &= \frac{j-1}{j}C^{j-1}(i) \\ &\quad + \frac{j-1}{j}\left(m^{j-1}(i) - u^j(i)\right)\left(m^{j-1}(i) - u^j(i)\right)^T \\ &\quad + \frac{1}{j}\left(\xi^j(i) - u^j(i)\right)\left(\xi^j(i) - u^j(i)\right)^T \end{aligned} \tag{10}$$

resulting in $\{u^j(i), Q^j(i)\}_{i=1}^{N}$.

Describing the whole body motions with respect to a latent space of reduced dimensionality instead of the joint space of high dimensionality is more convenient. The latent space changes every time that new teaching data is acquired. After the j-th kinesthetic teaching, $\{u^j(i), Q^j(i)\}_{i=1}^{N}$ is computed from (10) and the time and spatial components are separated: $u^j(i) = [u_t^j(i)\ u_s^j(i)]^T$

and $Q^j(i)=\begin{bmatrix} Q^j_{tt}(i) & Q^j_{st}(i) \\ Q^j_{ts}(i) & Q^j_{ss}(i) \end{bmatrix}$. Next, the mean v^j_s and the covariance matrix R^j_{ss} in the joint space are computed as follows:

$$v^j_s = \frac{1}{N}\sum_{i=1}^{N} u^j_s(i),$$
$$R^j_{ss} = \frac{1}{N}\sum_{i=1}^{N} Q^j_{ss}(i) + \frac{1}{N}\sum_{i=1}^{N}\left(u^j_s(i)-v^j_s\right)\left(u^j_s(i)-v^j_s\right)^T. \tag{11}$$

The eigenvectors and eigenvalues λ_i of the real symmetric covariance matrix R^j_{ss} are easily computed using PCA. The number of eigenvectors to construct the reduced dimensional latent space while representing the original dataset sufficiently is sought. A minimal integer q that satisfies $\frac{\Sigma_{i=1}^{q}\lambda_i}{\Sigma_{i=1}^{d}\lambda_i} > c$ is selected to indicate that the projection to the latent space covers 100% of the data's distribution. The first to the q-th eigenvectors construct a projection matrix W^j of size $d\times q$. Next, the projection matrix, including the temporal axis, is defined as

$$P^j = \begin{bmatrix} 1 & 0 \\ 0 & W^j \end{bmatrix}. \tag{12}$$

The new latent space L^j is defined as $L^j = \{v^j, P^j\}$, where $v^j = [0\ v^j_s]^T$. $\{u^j(i), Q^j(i)\}_{i=1}^N$ can be transformed onto the newly updated latent space L^j and expressed by $\{w^j(i), T^j(i)\}_{i=1}^N$ where

$$w^j(i) = P^{jT}(u^j(i)-v^j),$$
$$T^j(i) = P^{jT}Q^j(i)P^j. \tag{13}$$

2.5. Model parameter estimation

Previous investigations proposed selecting an optimal number of components K by minimizing the Bayesian information criterion (BIC) score, which reflects both model performance and complexity [11,12], using:

$$S_{BIC} = -\mathcal{L} + \frac{n_p}{2}\log\left(N_{tot}\right), \tag{14}$$

where n_p denotes the total number of model free parameters, N_{tot} the total number of data points, and $\mathcal{L}$ is the log-likelihood of the model described in (1).

$$\mathcal{L} = \sum_{i=1}^{N_{tot}}\log\left(p\left(x(i)\right)\right)$$
$$= \sum_{i=1}^{N_{tot}}\log\left(\sum_{k=1}^{K}p\left(k\right)p\left(x(i)\mid k\right)\right).$$

The number of model parameters is equal to

$$n_p = (K-1) + K\left(q + \frac{1}{2}q(q+1)\right). \tag{15}$$

The first term indicates the total number of prior probabilities, and the second term indicates the total number of means and elements in the symmetric covariance matrices.

The BIC score can be obtained after the full data are available. However, we want to know the number of Gaussian components K in the GMM during learning. Suppose that the j-th kinesthetic teaching is currently encoded, and a robot stores the whole data from the first to the j-th teachings in memory. Let total encoded data points so far in the latent space be expressed by $X = \left\{\{x^l(i)\}_{i=1}^N\right\}_{l=1}^j$. The data points can be expressed by K groups as $Y = \{Y_k\}_{k=1}^K = \left\{\{y_k(i)\}_{i=1}^{N_k}\right\}_{k=1}^K$ where N_k indicates the number of sample points in the group Y_k. Each group represents data points governed dominantly by a Gaussian component in the GMM.

Using the grouping, the GMM is set as follows.

$$G = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K,$$

where $\pi_k = \frac{N_k}{jN}$, $\mu_k = \frac{1}{N_k}\sum_{i=1}^{N_k} y_k(i)$, and $\Sigma_k = \frac{1}{N_k}\sum_{i=1}^{N_k}$ $\left(y_k(i)-\mu_k\right)\left(y_k(i)-\mu_k\right)^T$.

The expected value of $\mathcal{L}$ is computed by

$$E\left[\mathcal{L}\right] = E\left[\sum_{l=1}^{j}\sum_{i=1}^{N}\log\left(\sum_{k=1}^{K}\pi_k p\left(x^l(i)\mid k\right)\right)\right].$$

It is assumed that $p(y_k(i)\mid k) \gg p(y_k(i)\mid k^c)$, and $k^c \in \{1,2,\ldots,K\}-\{k\}$ indicates any other number but k. The assumption is taken based on the propensity that each Gaussian density component is dominant over partial data points where the data points tend to be linearly spread in a local space. Hence,

$$\sum_{l=1}^{j}\sum_{i=1}^{N}\log\left(\sum_{k=1}^{K}\pi_k p\left(x^l(i)\mid k\right)\right) \approx \sum_{k=1}^{K}\sum_{i=1}^{N_k}\log\left(\pi_k p\left(y_k(i)\mid k\right)\right),$$

because

$$E\left[\mathcal{L}\right] \approx \sum_{k=1}^{K}\sum_{i=1}^{N_k}E\left[\log\left(\pi_k p\left(y_k(i)\mid k\right)\right)\right]. \tag{16}$$

$$\sum_{k=1}^{K}\sum_{i=1}^{N_k}E\left[\log\left(\pi_k p\left(y_k(i)\mid k\right)\right)\right]$$
$$= \sum_{k=1}^{K}\sum_{i=1}^{N_k}E\left[\left(-\frac{1}{2}(y_k(i)-\mu_k)^T\Sigma_k^{-1}(y_k(i)-\mu_k)\right)\right]$$
$$+\sum_{k=1}^{K}N_k\left(\log(\pi_k) - \frac{1}{2}q\log(2\pi) - \frac{1}{2}\log\left(\left|\Sigma_k\right|\right)\right)$$
$$= \sum_{k=1}^{K}N_k\left(\log(\pi_k) - \frac{1}{2}q - \frac{1}{2}q\log(2\pi) - \frac{1}{2}\log\left(\left|\Sigma_k\right|\right)\right),$$

because

$$\sum_{k=1}^{K}\sum_{i=1}^{N_k}E\left[\left(-\frac{1}{2}\left(y_k(i)-\mu_k\right)^T\Sigma_k^{-1}\left(y_k(i)-\mu_k\right)\right)\right]$$
$$=\sum_{k=1}^{K}E\left[\sum_{i=1}^{N_k}\left(-\frac{1}{2}(y_k(i)-\mu_k)^T\Sigma_k^{-1}(y_k(i)-\mu_k)\right)\right]$$
$$=-\sum_{k=1}^{K}\frac{1}{2}qN_k=-\frac{1}{2}qjN.$$

This result leads to the following approximation:

$$E[\mathcal{L}]=-\frac{1}{2}qjN\left(1+\log(2\pi)\right)+\sum_{k=1}^{K}N_k\left(\log(\pi_k)-\frac{1}{2}\log\left(|\Sigma_k|\right)\right). \quad (17)$$

Through the procedure from Section 2.3 to 2.4, the integrated trajectory parameters, $\{w^j(i),T^j(i)\}_{i=1}^N$ are generated (see (13)).

To assign an initial GMM, every two successive data points is grouped. The initial condition aims to avoid zero variance along the time axis. Thus, N_k is $2j$, and K is $N/2$ initially. Then, an initial GMM can be declared, using $\{w^j(i),T^j(i)\}_{i=1}^N$, with

$$\begin{aligned}\pi_k&=\frac{2}{N}\\ \mu_k&=\frac{1}{2}\left(w^j(2k-1)+w^j(2k)\right)\\ \Sigma_k&=\frac{1}{2}\left(T^j(2k-1)+\left(w^j(2k-1)-\mu_k\right)\left(w^j(2k-1)-\mu_k\right)^T\right)\\&\quad+\frac{1}{2}\left(T^j(2k)+\left(w^j(2k)-\mu_k\right)\left(w^j(2k)-\mu_k\right)^T\right).\end{aligned} \quad (18)$$

Equation (18) does not require data points explicitly. By considering possible merges of sequential groups repeatedly, the correct number of K is found. Let $\rho_k=N_k\left(\log(\pi_k)-\frac{1}{2}\log\left(|\Sigma_k|\right)\right)$. While two k-th and $(k+1)$-th components are merged into a component (the new k-th component), the fitting term value is changed as much as:

$$\begin{aligned}E[\Delta\mathcal{L}]&=E[\rho_k^{'}-(\rho_k+\rho_{k+1})]\\&=N_k^{'}\left(\log\left(\pi_k^{'}\right)-\frac{1}{2}\log\left(\left|\Sigma_k^{'}\right|\right)\right)\\&\quad-N_k\left(\log\left(\pi_k\right)-\frac{1}{2}\log\left(|\Sigma_k|\right)\right)\\&\quad-N_{k+1}\left(\log\left(\pi_{k+1}\right)-\frac{1}{2}\log\left(|\Sigma_{k+1}|\right)\right)\\&=N_k^{'}\log\frac{\pi_k^{'}}{\sqrt{\left|\Sigma_k^{'}\right|}}-N_k\log\frac{\pi_k}{\sqrt{|\Sigma_k|}}-N_{k+1}\log\frac{\pi_{k+1}}{\sqrt{|\Sigma_{k+1}|}},\end{aligned} \quad (19)$$

where $\rho_k^{'}=N_k^{'}\left(\log\left(\pi_k^{'}\right)-\frac{1}{2}\log\left(\left|\Sigma_k^{'}\right|\right)\right)$ from $G=\{\pi_k^{'},\mu_k^{'},\Sigma_k^{'}\}_{k=1}^{K-1}$ after merging:

$$\begin{aligned}\pi_k^{'}&=\pi_k+\pi_{k+1},\\ \mu_k^{'}&=\frac{1}{\pi_k+\pi_{k+1}}\left(\pi_k\mu_k+\pi_{k+1}\mu_{k+1}\right),\\ \Sigma_k^{'}&=\frac{\pi_k}{\pi_k+\pi_{k+1}}\left(\Sigma_k+\left(\mu_k-\mu_k^{'}\right)\left(\mu_k-\mu_k^{'}\right)^T\right)\\&\quad+\frac{\pi_{k+1}}{\pi_k+\pi_{k+1}}\left(\Sigma_{k+1}+\left(\mu_{k+1}-\mu_k^{'}\right)\left(\mu_{k+1}-\mu_k^{'}\right)^T\right).\end{aligned}$$

$N_k^{'}=N_k+N_{k+1}$ is the number of sample points in the new k-th component.

The complexity term value also changes from (15), as much as

$$\Delta n_p=q+\frac{1}{2}q(q+1)+1. \quad (20)$$

Among all possible combinations of two sequential groups, a combination resulting in the largest reduction of $\Delta S_{BIC}=-\Delta\mathcal{L}+\Delta n_p$, as well as a negative ΔS_{BIC}, is selected. This process is repeated until the sign is no longer negative. After the termination, the number of remaining components is chosen as K. Furthermore, its corresponding model parameter set G is determined.

As notably documented by (19-20), the computation of ΔS_{BIC} relies on model parameters only. Therefore, no memorization of data points is required.

2.6. Model parameter refinement

Once K is selected, the expectation-maximization (EM) algorithm can be applied to refine the model parameters [18]. $\{w^j(i),T^j(i)\}_{i=1}^N$ generalizes the data information so far. An α number of random samples are stochastically generated from each distribution of $\mathcal{N}(z;w^j(i),T^j(i))$ and represented by $z^j=\{z^j(i)\}_{i=1}^{\alpha N}$ to run the EM. Using the samples, the EM is operated until convergence:

0) The model parameters form Section 2.5.

1) Compute $p_k^{t+1}(i)=\dfrac{\pi_k^t\mathcal{N}(z^j(i);\mu_k^t,\Sigma_k^t)}{\sum_{l=1}^{K}\pi_l^t\mathcal{N}(z^j(i);\mu_l^t,\Sigma_l^t)}$.

2) Compute

$$\begin{aligned}\pi_k^{t+1}&=\frac{1}{\alpha N}\sum_{i=1}^{\alpha N}p_k^{t+1}(i),\\ \mu_k^{t+1}&=\frac{1}{\alpha N\pi_k^{t+1}}\sum_{i=1}^{\alpha N}p_k^{t+1}(i)z^j(i),\\ \Sigma_k^{t+1}&=\frac{1}{\alpha N\pi_k^{t+1}}\sum_{i=1}^{\alpha N}p_k^{t+1}(i)\left(z^j(i)-\mu_k^{t+1}\right)\left(z^j(i)-\mu_k^{t+1}\right)^T.\end{aligned}$$

(t is the iteration number)

Iterate 1 and 2 until convergence $\left|\dfrac{\mathcal{L}^{t+1}}{\mathcal{L}^t}-1\right|<C_2\ll 1$ where C_2 is a iteration escape threshold and $\mathcal{L}=$

$\sum_{i=1}^{N} \log\left(p(z^j(i))\right).$

2.7. Motion generation

A generalized motion $\{\mu^j(t(i))\}_{i=1}^N$ is obtained through (2) and (3) using the converged parameter values from the EM. To produce the generalized motion $\{\phi^j(i)\}_{i=1}^N$ in the joint space, the projection to the joint space is applied as

$$\phi^j(i) = P^j \mu^j(t(i)) + v^j. \tag{21}$$

The motion $\phi^j = \{\phi^j(i)\}_{i=1}^N$ represents a learned motion from j kinesthetic teachings to date.

3. MOTION PRODUCTION FROM LEARNED MOTIONS

This section explains a method for a robot to produce a new motion through the of learned motions. Imagine that a human teaches a motion to a robot kinesthetically by moving the robot's limb. In the case of a complicated motion, it is difficult for the human teacher to move all robotic limbs simultaneously. Instead, it may be better for the human teacher to move limbs partially by dividing the whole motion into pieces. Then, the robot can integrate the partial limb motions into one whole motion. A new motion is reproduced from learned motions without new kinesthetic teaching sessions through the spatial combination of two learned motions. Furthermore, the time-shifted spatial combination makes more complex motion generation possible.

3.1. Spatial fusion of learned motions

When two motions are learned kinesthetically, it is possible to produce a new motion through the fusion of the two motions in the joint space when the two motions have the same temporal duration. Certain joint motions are selected from the first motion, and the other motions are selected from the second motion by a teacher. The two learned motions are described by different motion models. To fuse them, a new motion model is computed as follows.

A diagonal matrix D_1 is introduced with a i-th diagonal element that is one if the i-th joint component is selected from the first motion and zero otherwise. Similarly, a diagonal matrix D_2 can be assigned to indicate the joint selection from the second motion, and it should be true that $D_2 = I - D_1$.

Assuming that the two motions are learned from the same number of teaching trials, their fused motion can be expected to be learned from virtual trials that are expressed by

$$\xi_s^{(3)j}(i) = D_1 \xi_s^{(1)j}(i) + D_2 \xi_s^{(2)j}(i), \tag{22}$$

where $\{\xi_s^{(1)j}(i)\}_{i=1}^N$ and $\{\xi_s^{(2)j}(i)\}_{i=1}^N$ are the aligned posture vector data from the first and second motion teachings, respectively, and $\{\xi_s^{(3)j}(i)\}_{i=1}^N$ is the posture vector data that represent the virtual teaching of the fused motion.

However, the posture vector data are not available simultaneously. Furthermore, the robot does not store them but model parameters. The GMM of the first motion is in a different latent space from that of the second motion, and they generally have different numbers of Gaussian components. Therefore, they cannot be fused directly. Instead, the GMR in the joint space is computed from (3). Suppose the GMMs of the first and second motions in the latent spaces are $\{m^{(1)}(i), C^{(1)}(i)\}_{i=1}^N$ and $\{m^{(2)}(i), C^{(2)}(i)\}_{i=1}^N$, respectively. Thus,

$$m^{(1)}(i) = P^{(1)} \mu^{(1)}(t(i)) + v^{(1)},$$
$$C^{(1)}(i) = P^{(1)} \Sigma^{(1)}(t(i)) P^{(1)^T},$$

and

$$m^{(2)}(i) = P^{(2)} \mu^{(2)}(t(i)) + v^{(2)},$$
$$C^{(2)}(i) = P^{(2)} \Sigma^{(2)}(t(i)) P^{(2)^T},$$

where each latent space information is given.

$\{m^{(1)}(i), C^{(1)}(i)\}_{i=1}^N$ and $\{m^{(2)}(i), C^{(2)}(i)\}_{i=1}^N$ represent parameters obtained through the GMR for the first and the second motions, respectively. Each expression includes temporal and spatial components, such as $m^{(j)}(i) = \left[m_t^{(j)}(i)\ m_s^{(j)}(i)\right]^T$ and $C^{(j)}(i) = \begin{bmatrix} 0 & 0 \\ 0 & C_{ss}^{(j)}(i) \end{bmatrix}$, $j = 1, 2, 3$. The parameters are with respect to the joint space; therefore, the following fusion can be fulfilled and induced from (22):

$$\begin{aligned} m_s^{(3)}(i) &= D_1 m_s^{(1)}(i) + D_2 m_s^{(2)}(i), \\ C_{ss}^{(3)}(i) &= D_1 C_{ss}^{(1)}(i) D_1^T + D_2 C_{ss}^{(2)}(i) D_2^T, \\ m_t^{(1)}(i) &= m_t^{(2)}(i) = m_t^{(3)}(i). \end{aligned} \tag{23}$$

Using a new set of parameters, $\{m^{(3)}(i), C^{(3)}(i)\}_{i=1}^N$, the learning procedure from Section 2.4 to 2.6 results in a new produced motion model.

3.2. Spatial fusion of learned motions with time-shift

In the previous section, a new motion is produced by combining different local limb motions spatially. This approach could be further extended by combining different local limb motions from different time intervals. For example, suppose that two motions exist: a left arm motion for the first 5 seconds from the first motion, and a low limb motion for the last 8 seconds from the second motion. A motion can begin with any time-shift relative to the other motion. It is more natural that the two motions have different time lengths and do not begin at the same time.

Let the two motions have different time lengths. With the same time step for sampling regression parameters,

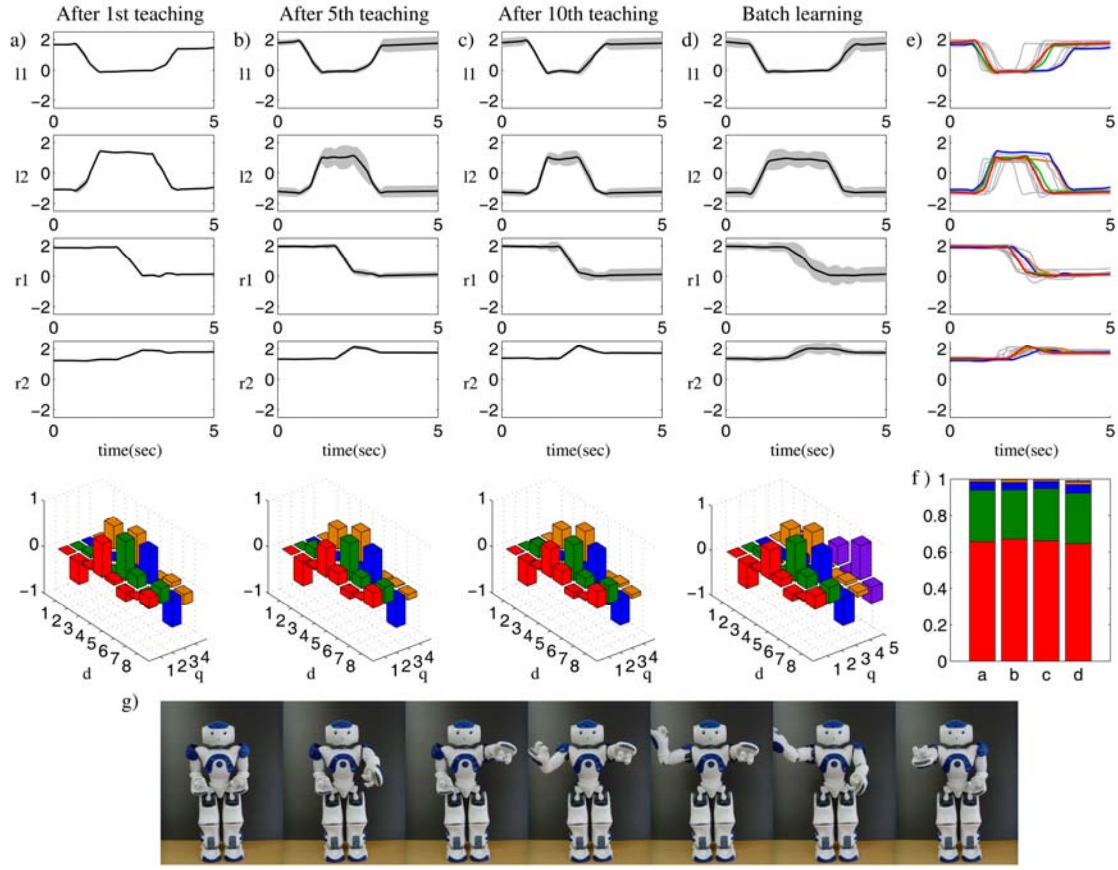

Fig. 2. Incremental learning of a martial gesture over the ten teaching sessions. Selected joint trajectories in radians are illustrated. l1 and l2 indicate two joints in the left arm and r1 and r2 in the right arm. (a)-(c) represent the learned joint trajectories after the first, the fifth, and the tenth teaching sessions, respectively, using the proposed learning algorithm; (d) represents the learned joint trajectories through the batch learning algorithm (see texts), and (e) shows them together with ten teaching trajectories. In the bottom of (a)-(d), major principal directional components (eigenvectors) are plotted correspondingly. For clarity, only eight element values are shown: the first to the fourth principal direction (q=4) of the latent space in (a)-(c) and the fifth principal direction (q=5) in (d). (f) illustrates the covering portions of the principal components in (a)-(d). (g) shows the snapshots of the gesture generated by a robot using the learned trajectories in (c).

they have different numbers of samples $N^{(1)}$ and $N^{(2)}$. For the two motions, suppose model parameters in the joint space are $\{m^{(1)}(i), C^{(1)}(i)\}_{i=1}^{N^{(1)}}$ and $\{m^{(2)}(i), C^{(2)}(i)\}_{i=1}^{N^{(2)}}$, respectively. To generate the second motion in t_d after the first motion begins, we merge the two motions spatially only during their overlapping period. Therefore,

$$m_s^{(3)}(i) = \begin{cases} m_s^{(1)}(i) & i \in Z^{(1)} - Z^{(2)} \\ D_1 m_s^{(1)}(i) + D_2 m_s^{(2)}(i - i_d) & i \in Z^{(1)} \cap Z^{(2)} \\ m_s^{(2)}(i - i_d) & i \in Z^{(2)} - Z^{(1)}, \end{cases}$$

$$C_{ss}^{(3)}(i) = \begin{cases} C_{ss}^{(1)}(i) & i \in Z^{(1)} - Z^{(2)} \\ D_1 C_{ss}^{(1)}(i) D_1^T + D_2 C_{ss}^{(2)}(i - i_d) D_2^T & i \in Z^{(1)} \cap Z^{(2)} \\ C_{ss}^{(2)}(i - i_d) & i \in Z^{(2)} - Z^{(1)}, \end{cases}$$

$$m_t^{(3)}(i) = m_t^{(1)}(i) = m_t^{(2)}(i - i_d) + t_d,$$

where $Z^{(1)} = \{0, \ldots, N^{(1)}\}$ and $Z^{(2)} = \{i_d, \ldots, N^{(2)} + i_d\}$, and i_d is the number of samples for the second motion after t_d.

By using the parameter set, the learning procedure from Section 2.4 to 2.6 produces a motion model of the merged motion.

4. EXPERIMENTS

The experiments in this paper use a Nao humanoid robot (Aldebaran Inc.) with 25 degrees of freedom (DOFs) as a test-bed. Each robot motor is set passively. In each kinesthetic teaching session, joint angle values are sequentially encoded at a rate of 20 Hz.

4.1. Incremental motion learning

A human taught a martial art gesture (hand blade hitting) to the Nao robot by moving its limbs at each kinesthetic teaching session through the proposed learning algorithm. Fig. 2 illustrates the evolution of its learned motion in radians over teaching sessions. Fig. 2(a)-(c) show reproduced trajectories of selected joints and major principal directional components of the latent space after the first, the fifth, and the tenth teaching sessions, respectively. Fig. 2(d) shows the trajectories obtained through the batch learning algorithm proposed in [11] for comparison. In the batch case, the ten teaching data are used at once for learning. To make the batch algorithm comparable with ours, the DTW is applied to raw teaching data trajectories to align them temporally. The DTW requires a motion template. In this case, the first teaching trajectories are assigned as template trajectories. An optimal number of K is selected for the batch algorithm as explained in [11,12]. To build latent spaces, c=0.99 is chosen. The proposed learning algorithm results in four principal directions of latent spaces (q=4) over the teaching sessions, while the batch algorithm results in five principal directions of latent space (q=5). The proposed learning algorithm is incremental; therefore, lost information is not recovered. This loss of information may explain why the number of principal directions is different from the batch case. However, the first four principal directions are fairly similar between the two cases as seen in the bottom of (a)-(d) where the major principal directional components of each latent space are plotted. Only eight element values are shown for clarity. Elements from the first to the fifth principal directional components are red-, green-, blue-, orange-, and violet-colored, respectively. Furthermore, the influence of the fifth principal direction in the batch case is negligible, as shown in (f) where the ratios of principal directional motion spreads $\left(\frac{\lambda_i}{\sum_{i=1}^{d}\lambda_i}\right)$ from (a)-(d) are illustrated. Motion spreads over the first to the fourth principal directions in (c)-(d) are fairly comparable. (e) overlaps the four reproduced trajectories from (a)-(d) for comparison. The blue, green, and red lines indicate the reproduced trajectories after the first, the fifth, and the tenth teaching sessions through the proposed learning algorithm, respectively. The orange line shows the reproduced trajectories through the batch learning algorithm. The gray lines represent encoded raw teaching data trajectories. Table 1 indicates that q does not change over teaching sessions, probably because the teaching motions are similar. Also, K converges quickly, which implies the learning was rapid. In fact, Fig. 2(b) shows almost the same joint trajectories and latent space components as in Fig. 2(c). Fig. 2(g) illustrates the sequential snapshots of the learned gesture performed by the robot after the tenth teaching. To further analyze the data, the batch learning algorithm is run repeatedly with different template trajectories. The first, fifth, and tenth teaching trajectories are selected as template trajectories.

In Fig. 3, for the three cases, mean square errors between reproduced and teaching trajectories are expressed over teaching sessions with different figures (◊: using the first teaching, △: using the fifth teaching, □: using the tenth teaching). Each batch learning is implemented with teaching data up to the most recent teaching session. The mean square errors (represented by ○) of the proposed learning algorithm are plotted together. The temporal alignment of the proposed learning algorithm is incrementally modified to best fit the most recent data. Therefore, its errors are lower compared to the batch case. The experimental results demonstrate that the Nao robot learns motion well using the proposed incremental learning algorithm and available teaching data, and its learned motion includes essential characteristics of the gesture.

Table 1. The number of Gaussian components and the dimensionality of the latent space over teaching sessions during the learning of a martial gesture in Fig. 3 (c=0.99).

session	1	2	3	4	5	6	7	8	9	10
K	13	10	9	9	9	9	9	9	9	9
q	4	4	4	4	4	4	4	4	4	4

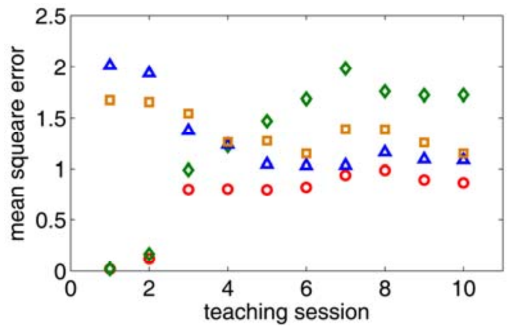

Fig. 3. Comparison of mean square errors between the reproduced and the teaching trajectories from each teaching session. The circle represents the case using the proposed learning algorithm and the others using the batch algorithms with different template trajectories for time alignment.

4.2. Motion production from learned motions

The Nao robot has learned three motions (left arm raising, right arm raising, and sit-down) as shown in Figs. 4(a, b, c), respectively, through the motion learning algorithm. Figs. 5(a, b, c) illustrate the motions performed by the robot. During the left arm motion execution, the robot does not move the right arm, and vice versa. Each of the learned motions in (a), (b), and (c) has 9, 10, and 7 Gaussian components (K) in the model, respectively. The dimensionality of the latent space (q) for (a), (b), and (c) are 3, 4, and 5, respectively. It is not easy for a human to move all the robot's limbs simultaneously during teaching. It is easier for, the human to teach the robot partial limb motions and subsequently let the robot integrate them spatially together. Fig. 4(d) shows the spatially fused motion of selected right and left arm motions through the proposed motion production method. In addition, an example of the spatial fusion with time-shift is demonstrated in Fig. 4(e). Arm motions are shifted 1.3 sec relative to leg motions during spatial integration. In Fig. 4(d), K is 12 and q is 5 for the fused motion. In Fig. 4(e), K is 13 and q is 6. Both K and q are scaled up compared to the K and q of the base motion. In aggregate, however, the space for the incremental algorithm is much smaller than the space for the base motion. Sequential snapshots in Figs. 5(d, e) verify the new motions per-

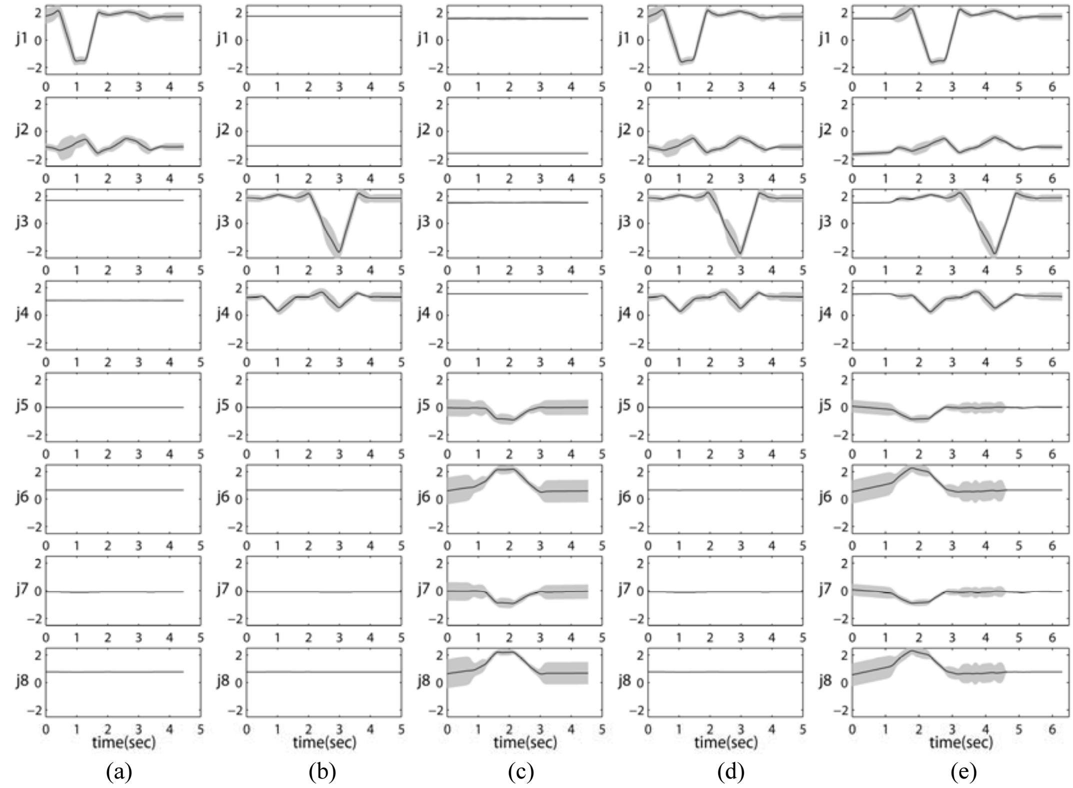

Fig. 4. New motion productions from pre-learned motions. Plots j1 and j2 indicate selected joints in the left arm, j3 and j4 in the right arm, j5 and j6 in the left leg, and j7 and j8 in the right leg. Plot series (a), (b), and (c) represent the motions prepared through the learning process. Plot series (d) and (e) represent newly produced motions from the motion production methods in Section 3.1 and 3.2, respectively. See the detail in the text. The unit for the y axis is radians.

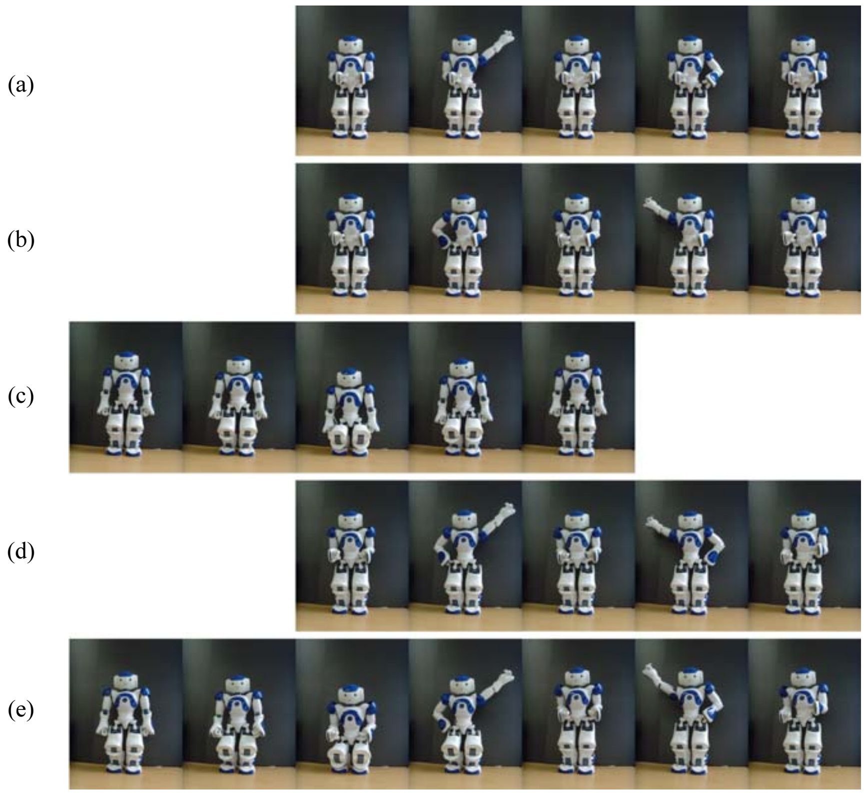

Fig. 5. Snapshots of motions generated by the robot. Each motion indicated by an alphabet from (a) to (g) corresponds to each plot indicated by the same letter in Fig. 4.

formed by the robot. The experimental results demonstrate the feasible performance of the proposed production algorithm.

5. CONCLUSIONS

This paper presented a learning approach for motions to produce a learned motion at any time. With further kinesthetic teachings, the learned motion becomes more refined. The approach is generally applicable for any learning by imitation scheme. From data encoding to motion generation, the overall procedure for the proposed approach is incremental. No preliminary information is required to learn a motion. During the process, information loss, especially due to dimensional reduction, is inevitable. However, experimental evaluations verify that the proposed approach can produce a generalized motion that reflects the essential characteristics of the teaching motions. In addition, a new motion production approach from learned motions without extra teaching sessions was proposed. A new motion can be generated by spatially combining learned motions without full learning. Time-shifted fusion is also implementable. Combining the two learning and production methods, a robot becomes able to produce a range of complex motions through accessible human interaction.

REFERENCES

[1] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol. 37, no. 2, pp. 286-298, 2007.

[2] A. P. Shon, J. J. Storz, and R. P. N. Rao, "Towards a real-time Bayesian imitation system for a humanoid robot," *Proc. Int Conf Robotics Automation*, Roma, Italy, 2007.

[3] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Trajectory formation for imitation with nonlinear dynamical systems," *Proc. IEEE Int. Conf. Intelligent Robots Systems*, pp. 752-757, 2001.

[4] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," *Proc. 14th Int. Conf Machine Learning (ICML)*, pp. 12-20, July 1997.

[5] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," *Proc. Int. Conf. Robotics Automation*, Kobe, Japan, 2009.

[6] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends in Cognitive Sciences*, vol. 6, no. 11, pp. 481-487, 2002.

[7] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Robot Programming by Demonstration*, B. Siciliano, and O. Khatib (eds.), Handbook of Robotics, pp. 1371-1394, Springer, 2008.

[8] B. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469-483, May 2009.

[9] S. Schaal, "Is imitation learning the route to humanoid robots," *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233-242, 1999.

[10] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamic system modulation for robot learning via kinesthetic demonstrations," *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1463-1467, 2008.

[11] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," *Proc. IEEE Int. Conf. Human Robot Interaction*, Arlington, Virginia, USA, pp. 255-262, 2007.

[12] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM," *Proc. 22nd Int. Conf. Mach. Learn.*, pp. 105-112, 2005.

[13] J. Kwon and F.C. Park, "Natural movement generation using hidden Markov models and principal components," IEEE *Trans. Syst., Man, Cybern. B, Cybern*, vol. 38, no. 5, pp. 1184-1194, Oct. 2008.

[14] D. B. Grimes, D. R. Rashid, and R. P. N. Rao, "Learning nonparametric models for probabilistic imitation," *Proc. Adv Neural Inf Process Syst.*, vol. 19, pp. 521-528, 2006.

[15] C. Wan and L. Liu, "Research and improvement on embedded system application of DTW-based speech recognition," *Proc IEEE Int. Conf. ASID*, 2008.

[16] E. J. Keogh and M. J. Pazzani, "Derivative Dynamic Time Warping," *First SIAM Int. Conf. Data Mining*, Chicago, IL, 2001.

[17] E. Keogh and C. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, pp. 358-386, 2005.

[18] A. Dempster, A. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1-38, 1977.

Sumin Cho received his B.S. degree in the Department of Computer Science from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2010. He will soon complete an M.S. degree and pursue a Ph.D. degree in Computer Science from Korea Advanced Institute of Science and Technology, Daejeon, Korea. His research interests include human-robot interaction, and intelligent robotics.

Sungho Jo received his B.S. degree in the School of Mechanical & Aerospace Engineering from Seoul National University, Seoul, Korea, in 1999. He completed an M.S. degree in Mechanical Engineering and a Ph.D. degree in Electrical Engineering and Computer Science from Massachusetts Institute of Technology, Cambridge, MA, in 2001 and 2006, respectively. From 2006 to 2007, he was a Postdoctoral Researcher at the MIT Media Lab, Cambridge, MA. Since 2007, he has been an Assistant Professor in the Department of Computer Science at Korea Advanced Institute of Science and Technology, Daejeon, Korea. His research interests include human-robot interaction, biomimetic robots, and brain-machine interface.