Robotics and Autonomous Systems 75 (2016) 260–272

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Real-time Unsupervised Segmentation of human whole-body motion and its application to humanoid robot acquisition of motion symbols

Wataru Takano *, Yoshihiko Nakamura

Mechano-Informatics, University of Tokyo, 7-3-1, Hongo, Bunkyoku, Tokyo, 113-8656, Japan

HIGHLIGHTS

- This paper proposes a framework for real-time unsupervised segmentation of human motions and automatic symbolization of the motions.
- The segmentation is based on prediction uncertainty and symbolization is based on competitive learning of human motion.
- Their integration was verified on the human motion datasets.

ARTICLE INFO

Article history:
Received 17 February 2015
Received in revised form
25 July 2015
Accepted 26 September 2015
Available online 9 October 2015

Keywords:
Motion segmentation
Motion primitive
Competitive learning

ABSTRACT

An interactive loop between motion recognition and motion generation is a fundamental mechanism for humans and humanoid robots. We have been developing an intelligent framework for motion recognition and generation based on symbolizing motion primitives. The motion primitives are encoded into Hidden Markov Models (HMMs), which we call "motion symbols". However, to determine the motion primitives to use as training data for the HMMs, this framework requires a manual segmentation of human motions. Essentially, a humanoid robot is expected to participate in daily life and must learn many motion symbols to adapt to various situations. For this use, manual segmentation is cumbersome and impractical for humanoid robots. In this study, we propose a novel approach to segmentation, the Real-time Unsupervised Segmentation (RUS) method, which comprises three phases. In the first phase, short human movements are encoded into feature HMMs. Seamless human motion can be converted to a sequence of these feature HMMs. In the second phase, the causality between the feature HMMs is extracted. The causality data make it possible to predict movement from observation. In the third phase, movements having a large prediction uncertainty are designated as the boundaries of motion primitives. In this way, human whole-body motion can be segmented into a sequence of motion primitives. This paper also describes an application of RUS to AUtonomous Symbolization of motion primitives (AUS). Each derived motion primitive is classified into an HMM for a motion symbol, and parameters of the HMMs are optimized by using the motion primitives as training data in competitive learning. The HMMs are gradually optimized in such a way that the HMMs can abstract similar motion primitives. We tested the RUS and AUS frameworks on captured human whole-body motions and demonstrated the validity of the proposed framework.

1. Introduction

In robotics, various imitative learning frameworks have been proposed [1–3]. These approaches symbolize body movement into a set of model parameters, which are referred to as motion symbols, recognize observed motion as motion symbols, and generate motion data from the motion symbols [4–8]. This research on constructing intelligence through encoding bodily senses and movement into motion symbols in robotics has been inspired by the discovery of mirror neurons [9,10] and by the hypothesis of mimesis [11]. Mirror neurons fire not only when a macaque monkey observes another monkey performing a particular motion but also when the first monkey has just performed the same motion. The relationships between mirror neurons and various functions such as symbolization, recognition, generation of behaviors, communication, theory of mind, and language have attracted much attention. The mimesis hypothesis posits that intelligence in human beings originated in gesture communication, so that people would have gained the ability to memorize gestures performed by

* Corresponding author. Tel.: +81 3 5841 6378; fax: +81 3 3818 0835.
E-mail addresses: takano@ynl.t.u-tokyo.ac.jp (W. Takano),
nakamura@ynl.t.u-tokyo.ac.jp (Y. Nakamura).

http://dx.doi.org/10.1016/j.robot.2015.09.021

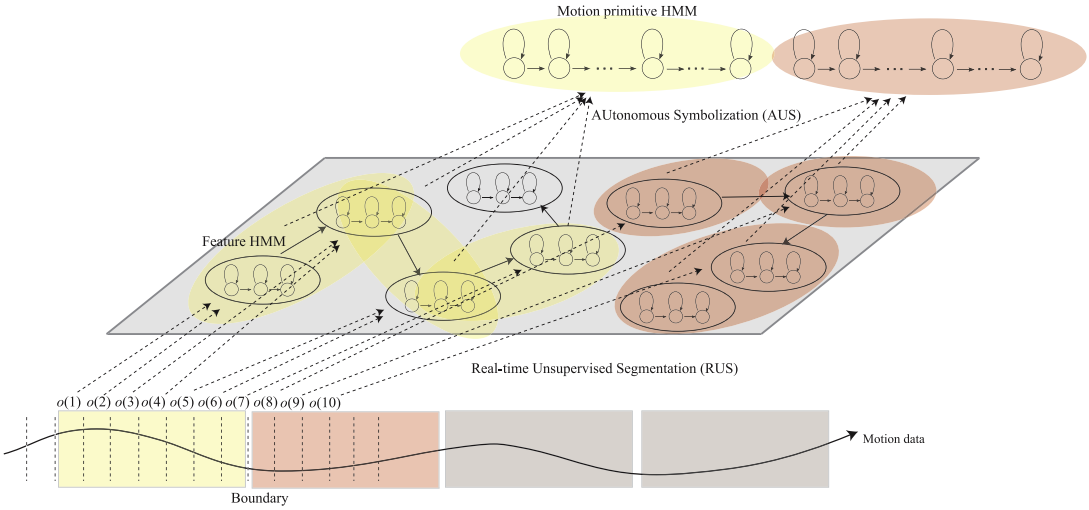


Fig. 1. Illustration of a hierarchical processes Real-time Unsupervised Segmentation (RUS) and AUtonomous Symbolization (AUS).

others, recognize behaviors, and synthesize behaviors before acquiring language.

When a humanoid robot memorizes human behavior as a motion symbol through observation, the robot needs to segment seamless human behavior into motion primitives and to encode these motion primitives into model parameters. That is, symbolization of behaviors should follow motion segmentation. In previous frameworks to symbolize human behaviors, motion primitive data must be provided by manually segmenting measured behaviors, such as captured human motions. Using manually selected motion primitives has the advantage that it is easy to give motion labels to them because they are intuitive. However, when a humanoid robot needs to memorize a large amount of motions in the form of motion symbols, manual segmentation is not practical because of the required labor. Additionally, a humanoid robot must be able to segment seamless human behavior by itself in order to incrementally develop motion symbols by encoding unknown motion primitives into new model parameters. Thus, for a humanoid robot that coexists in our daily lives, automatic segmentation of human behaviors is a fundamental intelligent processing that leads to memorizing, recognizing, and synthesizing behaviors based on the motion symbols.

In this study, we propose a novel approach to Real-time Unsupervised Segmentation (RUS) for human whole-body motions. RUS detects boundaries of motion primitives that are frequently observed in human behaviors. As illustrated by Fig. 1, in the first phase, human behavior is divided into a sequence of short feature movements that are encoded into Hidden Markov Models (HMMs), referred to as "feature HMMs". This phase converts continuous human behavior into a sequence of discrete features. In the second phase, causality among the discrete features is extracted from sequences of these features by a correlation matrix. In the third phase, the correlation matrix is used to estimate prediction uncertainty for the discrete feature that will follow the current sequence of discrete features. Large prediction uncertainty implies that the current movement is unpredictable, and unpredictable movement is identified as a boundary of motion primitives. In this way, motion primitives can be derived. Additionally, this paper applies RUS to AUtonomous Symbolization (AUS) of human whole-body motions. The derived motion primitives are used as training data for HMMs, which are then referred to as "motion symbols". Each motion primitive is classified as a motion symbol, and the corresponding HMM retrains the motion primitive incrementally using competitive learning. RUS and AUS allow a humanoid robot to observe human behaviors, derive motion segments, and acquire motion symbols by itself. We tested a framework integrating RUS and AUS on captured human behaviors and demonstrated its validity.

2. Related research

Segmentation has been studied from various points of view, including motion, speech, and sentence structure. Mori et al. developed a technique for supervised segmentation of daily human movements [12]. Two of their assumptions are that the boundaries of daily human motion segments are unclear and that motion boundaries are distributed. Human subjects are asked to evaluate, on a scale from one to four, the possibility that each frame in a sequence of human movements is a boundary. The evaluation scores are taken as the distribution of motion boundaries. Since the distribution is based on manual segmentation, criteria for segmentation similar to intuitive segmentation can be obtained. However, this approach has the drawback that it is not scalable to large motion datasets. As the number of motion datasets to be processed increases, the labor necessary for manual segmentation also increases. Kohlmorgen et al. proposed a system for automatic segmentation of time series data [13], and Kulic et al. or Janus et al. applied their system to incremental segmentation and clustering of human motion pattern primitives [14,15]. This segmentation algorithm assumes that same motion primitives have same underlying distribution. The window of motion data is represented by the Gaussian distribution, and the node with this distribution is added to an HMM. The node path corresponding to the motion data is estimated by Viterbi algorithm in the HMM, and the boundary of the motion primitive can be detected at the switching point between the end nodes in the path. The derived motion segments are placed into the closest group, and the large group forms multiple child groups. This incremental process results in the clustering structure of the motion primitives. The method of tuning several parameters in the segmentation and grouping is not described, which is critical to appropriate segmentation and clustering of motion primitives. Grave et al. developed an approach to segmenting and classifying the motion to manipulate an object. The likelihood of a motion segment terminating at a specific point being generated by an HMM and the likelihood of a motion segment starting at the same point begin generated by the following HMM are computed, and the point is searched for that maximizes the combined likelihood. The initial HMMs require a dataset of presegmented motion data [16].

An approach to extracting periodic motion primitives from a conductor's hands has also been proposed [17]. This approach is based on the COMPRESSIVE technique, which measures the compression rate of hand movements. The compression rate is computed from the length and frequency of motion chunks. A

motion with a high compression rate is regarded as a motion primitive. This method can segment periodic human movements into a sequence of motion primitives. Because this approach counts the occurrence of a specified motion chunk to compute its frequency by observation of human motion, the system has to store all observations simultaneously. Moreover, the extraction of motion primitives that have a high compression rate is time consuming. It is, therefore, difficult to apply this approach to segmenting long whole-body motions of humans in real time. Taehoon et al. [18] have proposed a method to segment rhythmic movement into motion primitives. Their method detects the moments when the velocity of each joint angle becomes zero and approximates a temporal sequence of these stopped moments by a sine function. The sine functions for all of the joints are superposed into a reference function for boundaries of motion primitives. Motion primitives can be extracted from rhythmic human movements by these reference functions. This method results in all of the derived motion primitives having the same interval because the reference function is represented by a periodic function.

Shiratori et al. also propose a method to derive motion segments from dance movement, based on an assumption that the dance movement is a sequence of motion primitives with boundaries at stopped moments [19]. To extract the motion primitives, the method focuses on the speed of a performer's hands and legs, selects a pose when one of the body parts stops as a candidate for a key pose, and chooses boundaries of motion primitives from among the candidates by taking into account the rhythm of the music. In this method, the dance movement is divided into motion primitives by assuming that the motion primitives start and end at frames with zero motion velocity. This assumption may be reasonable in dance situations, but it is not clear that this assumption is useful for daily human actions.

In the fields of natural language processing and speech recognition, research on segmentation has been conducted for a long time because segmentation of words from sentences or speech without clear boundaries between words is a fundamental process [20]. Language segmentation approaches developed thus far can be categorized into three strategies: utterance-boundary strategy [21,22], predictability strategy [23,24], and word-recognition strategy [25]. The utterance-boundary strategy hypothesizes that the ends of words have features similar to the ends of sentences or of utterances. The predictability strategy is based on predicting a phoneme or character by the immediately preceding phonemes or characters. The word-recognition strategy is an approach to checking whether a sequence of phonemes or characters matches one of a set of registered words. In recent years, the performance of computers has improved and connectionist models, such as neural networks, have been applied to the utterance-boundary strategy and the predictability strategy. Although these models make it possible to detect boundaries of words, it is still impossible to recognize a sequence of phonemes or characters as a word. Moreover, it is hard for the models to implement real-time learning, for which optimizing parameters of the models consumes a substantial amount of time. In contrast, the word-recognition strategy enables a sequence of phonemes or characters to be recognized as a word at the same time that segmentation is performed. However, a word-recognition strategy model that allows for real-time segmentation has not been proposed yet.

3. RUS of human whole-body motion

3.1. Encoding features of short movement

Human whole-body motion $\boldsymbol{O}$ is represented by a sequence of vectors of joint angles. Dividing human whole-body motion into short movements $\boldsymbol{o}(i)$ results in human motion being expressed as a sequence of short movements as shown by Fig. 2.

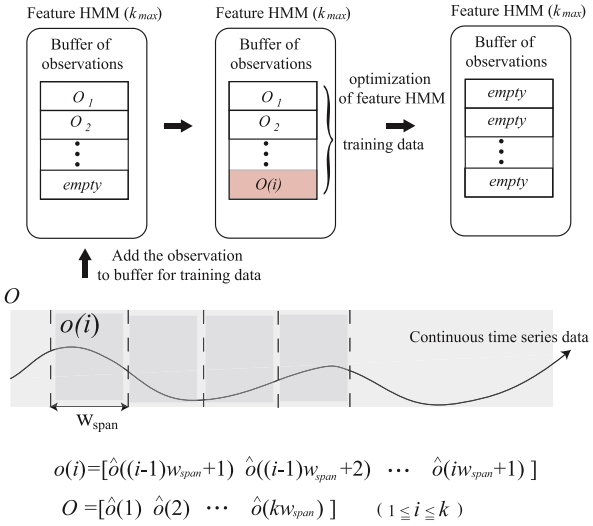


Fig. 2. Procedure for optimizing a feature HMM.

$$\boldsymbol{O} = [\boldsymbol{o}(1), \boldsymbol{o}(2), \ldots, \boldsymbol{o}(k)]\,. \tag{1}$$

A short movement is also represented by a short sequence of joint angle vectors at the tth frame, $\hat{\boldsymbol{o}}(t)$.

$$\boldsymbol{o}(i) = \left[\hat{\boldsymbol{o}}((i-1)w_{span}+1), \hat{\boldsymbol{o}}((i-1)w_{span}+2), \ldots \hat{\boldsymbol{o}}(iw_{span})\right] \tag{2}$$

where w_{span} is the number of frames in each short movement.

We introduce a set of N_D pieces of HMMs, into which the short movement $\boldsymbol{o}(i)$ is encoded. The HMMs corresponding to the short movement are referred to as a "feature HMM". HMM is a stochastic model that is used to categorize input data, especially in speech recognition. HMM is defined by a set of variables $\lambda = \{Q, A, B, \Pi\}$, where $Q = \{q_1, \ldots, q_n\}$ is a set of nodes, $A = \{a_{ij}\}$ is the matrix whose (i, j) element represents the transition probability from the ith node to the jth node, $B = \{b_1, b_2, \ldots, b_n\}$ is the set of probability density functions, and $\boldsymbol{\Pi} = \{\pi_1, \pi_2, \ldots, \pi_n\}$ is the set of initial node distributions. A probability density function is defined by the Gaussian distribution form

$$b_i(\hat{\boldsymbol{o}}) = \frac{1}{\sqrt{(2\pi)^m |\boldsymbol{\Sigma}_i|}} \exp\left\{-\frac{1}{2}(\hat{\boldsymbol{o}} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\hat{\boldsymbol{o}} - \boldsymbol{\mu}_i)\right\} \tag{3}$$

where $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and m denote the mean vector, the covariance matrix of the HMM, and the dimension of the input data, respectively.

We first calculate the likelihoods of a short movement being generated by feature HMMs and select the HMM $\lambda^f_{\mathcal{R}}$ with the largest likelihood. Let us denote the likelihood of a short movement $\boldsymbol{o}(i)$ being generated from the kth HMM λ^f_k by the conditional probability $P(\boldsymbol{o}(i)|\lambda^f_k)$. The HMM $\lambda^f_{\mathcal{R}}$ represents the HMM with the largest likelihood, $P(\boldsymbol{o}(i)|\lambda^f_k)$. Then,

$$\lambda^f_{\mathcal{R}} = \arg \max_{\lambda^f_k : k=1,2,\ldots,N_f} P(\boldsymbol{o}(i)|\lambda^f_k) \tag{4}$$

where N_f is the number of feature HMMs. As shown by Fig. 2, the short movement $\boldsymbol{o}(i)$ is provided for the HMM $\lambda^f_{\mathcal{R}}$ as training data such that the parameters of the HMM can be optimized by the Baum–Welch algorithm [26], which is an expectation–maximization algorithm. The optimizing procedures are iterated over a sequence of short movements. Note that the initial parameters of the feature HMMs are set randomly.

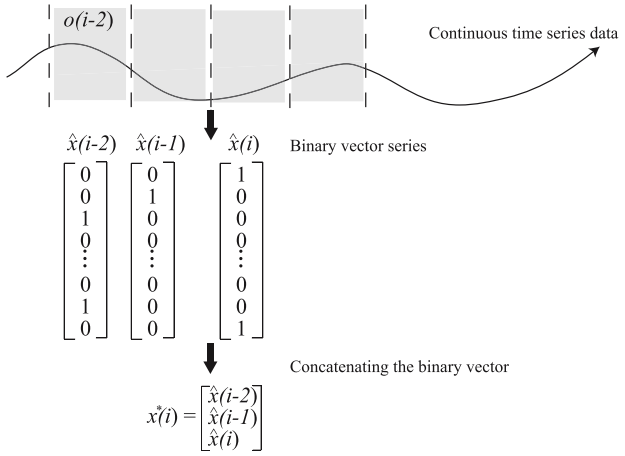


Fig. 3. A continuous motion data is converted to a sequence of binary vectors $\hat{\boldsymbol{x}}$. The binary vectors are concatenated into a feature vector $\boldsymbol{x}^*$.

3.2. Extracting causality among feature HMMs

A phase of extracting causality from among short movements follows the optimization of the feature HMMs. A short movement is recognized as the feature HMM with the largest likelihood that the movement is generated by that HMM. Thus, a sequence of short movements can be converted to a sequence of the feature HMMs corresponding to those movements. In this study, the N_s pieces of the feature HMMs with the largest likelihood are selected. The selection of several feature HMMs improves the robustness of the conversion from short movements to binary features. The set of the selected feature HMMs, $S(i)$, forms a binary feature vector corresponding to the short movement. The binary vector $\hat{\boldsymbol{x}}(i)$ is a column vector with N_f elements, each zero or one. Each element of $\hat{\boldsymbol{x}}(i)$ is set as follows.

$$\hat{\boldsymbol{x}}(i) = \left[\hat{x}_1(i)\, \hat{x}_2(i)\, \ldots\, \hat{x}_{N_f}(i)\right]^T \tag{5}$$

$$\hat{x}_k(i) = \begin{cases} 1 & \lambda_k^f \in S(i) \\ 0 & \lambda_k^f \notin S(i) \end{cases} \tag{6}$$

where ^{T} denotes matrix transposition.

Aligning the M pieces of the binary feature vectors in columns creates a feature vector $\boldsymbol{x}(i)$ that is a unit vector.

$$\boldsymbol{x}^*(i) = [\hat{\boldsymbol{x}}(i-M+1)^T \cdots \hat{\boldsymbol{x}}(i)^T]^T \tag{7}$$

$$\boldsymbol{x}(i) = \frac{\boldsymbol{x}^*(i)}{\|\boldsymbol{x}^*(i)\|}. \tag{8}$$

Fig. 3 illustrates an overview of conversion from the continuous motion data to the binary feature vector. The correlation matrix extracts the causality of the feature vectors from the sequence of feature vectors. Correlation learning is a scheme where the causality between an input vector and an output vector $\{\boldsymbol{u}_l, \boldsymbol{y}_l\ l = 1, 2, \ldots, K\}$ is represented by the correlation matrix $\boldsymbol{W}_0 = \sum_{l=1}^{K} \boldsymbol{y}_l \boldsymbol{u}_l^T$. In an ideal situation, where all input vectors are orthogonal to each other, each output vector $\boldsymbol{y}_k$ can be predicted from its corresponding input $\boldsymbol{u}_k$ as follows:

$$\boldsymbol{y}_k = \boldsymbol{W}_0 \boldsymbol{u}_k \tag{9}$$

because $\boldsymbol{u}_l^T \boldsymbol{u}_k = 1$ only if $l = k$.

The correlation matrix can be computed as described above if all pairs of input and output vectors are known in advance. In segmentation of human whole-body motion, however, it is preferable to determine the correlation matrix incrementally while a robot observes human motions rather than to learn it from the complete set of pairs of input and output vectors at once. Therefore, we propose an approach to incremental computation of the correlation matrix

$$\boldsymbol{W}(i) = \alpha \boldsymbol{W}(i-1) + \eta \boldsymbol{x}(i) \boldsymbol{x}(i-1)^T, \tag{10}$$

where α and η denote the stabilizing and learning coefficients, respectively. In this approach, a current feature vector $\boldsymbol{x}(i)$ is given as an output, and the feature vector $\boldsymbol{x}(i-1)$ immediately preceding $\boldsymbol{x}(i)$ is given as an input. The matrix $\boldsymbol{x}(i)\boldsymbol{x}(i-1)^T$ projects the preceding feature vector $\boldsymbol{x}(i-1)$ on to the current feature vector $\boldsymbol{x}(i)$. Eq. (10) gradually updates the correlation matrix from the previous matrix $\boldsymbol{W}(i-1)$ by incrementing the current dynamics $\boldsymbol{x}(i)\boldsymbol{x}(i-1)^T$ of the feature vectors. This correlation matrix can extract the causality of feature vectors and predict the feature vector one step ahead of the current feature vector. The norm of the predicted feature vector $\boldsymbol{W}\boldsymbol{x}$ is bounded below one by setting the stabilizing and learning coefficients so that the sum of them becomes one, $\alpha + \eta = 1$.

3.3. Segmentation of motion primitives

We intuitively suspect that we can easily predict the movement following the current observation at an intermediate time point within a motion primitive but that it is difficult to predict movement at the transition between two motion primitives. This intuition leads to the assumption that an error between actual movement and predicted movement can be used as a criterion for the boundary of a motion primitive. We can calculate the error $E(i)$ between the actual feature vector and the predicted feature vector as

$$E(i) = \|\boldsymbol{x}(i) - \boldsymbol{W}\boldsymbol{x}(i-1)\|. \tag{11}$$

The error can be interpreted as the prediction uncertainty, and the moments when this uncertainty is large will be considered the boundaries of motion primitives. In our implementation, we detect the boundaries as the moments when the error exceeds a specified threshold E_{th}. The time of the boundary for a motion primitive can be derived as k_B such that

$$E(k_B - 1) > E_{th}$$
$$E(k_B) < E_{th}.$$

4. Autonomous symbolization of human motion based on segmentation

A framework to symbolize human whole-body motions has been developed, where human motion primitives are encoded into their corresponding HMMs [6]. This framework enables a humanoid robot not only to memorize human motions as motion symbols but also to observe and generate human-like motions using motion symbols. Furthermore, motion data are decomposed into features typified by principal components or independent analysis, and the feature series are subsequently encoded into the HMM in order to reduce the dimensionality and noise in the original motion data [27]. The previous framework needs manual segmentation of seamless human motions in order to derive motion primitives, which are then given to the HMMs as training data. In this study, we integrate RUS with AUtonomous Symbolization of human motions (AUS). This integration makes it possible for a humanoid robot to autonomously acquire motion symbols from observation without manual intervention.

Symbolization of motion primitives by competitive learning happens after segmentation.

step1 Initially, a robot has N_S HMMs ($\lambda_k : k = 1, 2, 3, \ldots, N_S$). Randomly set the parameters of each HMM.

step2 Compute the likelihoods that derived motion primitive $\boldsymbol{O}_{segment}(i)$ is generated by the HMMs. Select the HMM with the largest likelihood, $\lambda_{\mathcal{R}}$, as the resultant motion primitive:

$$\lambda_{\mathcal{R}} = \arg \max_{\lambda_k : k=1,2,\ldots,N_S} P(\boldsymbol{O}_{segment}(i)|\lambda_k). \quad (12)$$

step3 Store the motion primitive $\boldsymbol{O}_{segment}(i)$ as training data for the HMM $\lambda_{\mathcal{R}}$.

step4 When the number of stored motion primitives for the training data of HMM $\lambda_{\mathcal{R}}$ reaches a specified quantity, optimize parameters of the HMM such that the likelihood of the motion primitives being generated by the HMM are maximized. After optimization, remove the motion primitives. When the number of stored motion primitives is less than the specified quantity, skip the optimization.

step5 Derive the next motion primitive $\boldsymbol{O}_{segment}(i+1)$ through the RUS method; then, go to *step2*.

The competitive learning algorithm incrementally optimizes the HMMs, and these HMMs gradually develop into motion symbols.

In the motion recognition process, the resultant motion primitive is the motion symbol with the largest likelihood, as computed by Eq. (12). In the motion generation process, the initial node $q(1)$ is selected according to the initial node probability $\boldsymbol{\Pi}$, and the joint angles $\boldsymbol{o}(1)$ at time $t = 1$ is subsequently sampled according to the output probability $b_{q(1)}(\boldsymbol{o})$. The node $q(2)$ at time $t = 2$ is selected according to the transition probabilities $a_{q(1)i}$ from node $q(1)$ to the ith node, and the joint angles $\boldsymbol{o}(2)$ are sampled according to the output probability $b_{q(2)}(\boldsymbol{o})$. These processes are iterated to generate a sequence of joint angles. This simple motion generation adopts a Monte-Carlo-based method [6].

5. Experiments on captured human whole-body motion

5.1. Segmentation of motion primitives

Experiments using the RUS and AUS frameworks were conducted on captured human whole-body motions. A human subject performing a seamless sequence of motion primitives was measured by an optical motion capture system. The positions of 34 markers attached to the subject were available, and an inverse kinematics computation converted these position data to a time series of 46-dimensional vectors. Each vector consists of 20 joint angles, vertical body position, roll, pitch angles, and their corresponding velocities [28]. Thus, human whole-body motion was represented by 46-dimensional vectors. Short movements $\boldsymbol{o}(i)$ were defined to consist of five frames of the vector series. The short movements were encoded into feature HMMs. We set the type of the HMM to left-to-right, the number of nodes to three, and the number of HMMs to 50 ($N_f = 50$). A feature vector $\boldsymbol{x}(i)$ was created by aligning four 50-dimensional binary vectors, each element of which corresponds to a feature HMM. The resultant feature vector $\boldsymbol{x}(i)$ was a 200-dimensional vector: $\boldsymbol{x}(i) = [\hat{\boldsymbol{x}}(i-3)^T, \hat{\boldsymbol{x}}(i-2)^T, \hat{\boldsymbol{x}}(i-1)^T, \hat{\boldsymbol{x}}(i)^T]^T$. The stabilizing and learning coefficients of the correlation matrix were set to 0.99 and 0.01 ($\alpha = 0.99$, $\eta = 0.01$).

In the first experiment, the human subject performed seven kinds of motion primitives: "left punch", "bend", "right kick", "left punch", "right punch", "left punch" and "bend". Four-minute human behavior was recorded where seven motion patterns were observed in random order. This recorded data was iteratively given to RUS and AUS as the training data, and they incrementally learned the boundaries of motion segments and subsequent motion primitives. Fig. 4 shows a sequence of snapshots for subject movements, time stamps for the boundaries of the motion primitives detected by RUS, and profiles of two joint angles and their velocities from the captured human motion. As shown by Fig. 4, a seamless human motion is appropriately segmented into motion primitives, each of which can be given one of the labels "left punch", "bend", "right kick", "left punch", "right punch", "left punch", or "bend". The profile of joint angular velocities shows that RUS does not segment the human behavior into motion primitives based on detection of a zero-velocity posture; that would have segmented the behavior into shorter motion primitives.

We measured the computational times for several processes in RUS. A computer with a 3.6 GHz Xeon processor was used for these measurements. The average times for using inverse kinematics to convert captured human data into the 46-dimensional vector, derivation of the feature vector $\boldsymbol{x}(i)$ after the inverse kinematics computation, and detection of the boundary from the feature vector were 7.3 (ms), 0.3 (ms), and 0.6 (ms), respectively. This result implies that RUS can be processed in real time since the sampling rate for motion capture was 30 (Hz). Additionally, the times required to train a feature HMM and to update a correlation matrix in the training phase were 6.3 (ms) and 0.8 (ms), respectively. These computational costs also validate RUS from a practical point of view.

We evaluated the validity of RUS by conducting a cluster analysis of the derived motion primitives. A total of 333 motion primitives were identified by RUS. Each motion primitive was encoded into its corresponding HMM. This procedure resulted in 333 HMMs. It is difficult to measure distances among the motion primitives directly. However, distances among the HMMs could be measured by use of Kullback Leibler information.

$$D^*(\lambda_i^p, \lambda_j^p) = \frac{1}{T_{Gi}} \ln P(\boldsymbol{O}_{Gi}|\lambda_i^p) - \ln P(\boldsymbol{O}_{Gi}|\lambda_j^p), \quad (13)$$

where λ_k^p ($k = 1, 2, 3, \ldots, 333$) is the HMM for the kth motion primitive, $\boldsymbol{O}_{Gk}$ is the motion primitive generated by the HMM λ_k^p, and T_{Gk} is the number of frames in the motion primitive $\boldsymbol{O}_{Gk}$. Kullback Leibler information $D^*(\lambda_i^p, \lambda_j^p)$ is asymmetric, but symmetry is obtained by Eq. (14)

$$D(\lambda_i^p, \lambda_j^p) = \frac{D^*(\lambda_i^p, \lambda_j^p) + D^*(\lambda_j^p, \lambda_i^p)}{2}. \quad (14)$$

Thus, the HMM-based distances $D(\lambda_i^p, \lambda_j^p)$ replace the actual distances among the motion primitives. The motion primitives are located in a multidimensional space such that the distance $d(x_i, x_j)$ between the ith and jth motion primitives in the space are as close as possible to the HMM-based distance $D(\lambda_i^p, \lambda_j^p)$ for each pair i, j. The distance $d(x_i, x_j)$ is measured as the Euclidean distance between the two locations x_i and x_j corresponding to the ith and jth motion primitives the space. A multidimensional scaling algorithm computes the location x_i for the ith motion primitives such that it can minimizes the following error function.

$$E = \frac{1}{{}_N C_2} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{(D(\lambda_i^p, \lambda_j^p)^2 - d(x_i, x_j)^2)^2}{4D(\lambda_i^p, \lambda_j^p)^2}. \quad (15)$$

Eq. (15) represents the error between the HMM-based distance and the Euclidean distance in the form of a four-dimensional polynomial; this expression makes it possible to compute the location x_i by the Newton–Raphson method.

Fig. 5 shows the relation between the number of dimensions of the space and the error given by Eq. (15). The relation reveals that the rate of change of the error, $\gamma = \frac{E_d - E_{d+1}}{E_d}$, becomes zero in 4-dimensional space (E_d is the distance error of the d-dimensional space). Therefore, we constructed a 4-dimensional space where the motion primitives were distributed. Fig. 6 shows the constructed space: each point represents each motion primitive; the color and shape of the point signifies the categorization of the motion primitive. Each motion primitive is categorized according to motion symbols. The motion symbols are autonomously acquired,

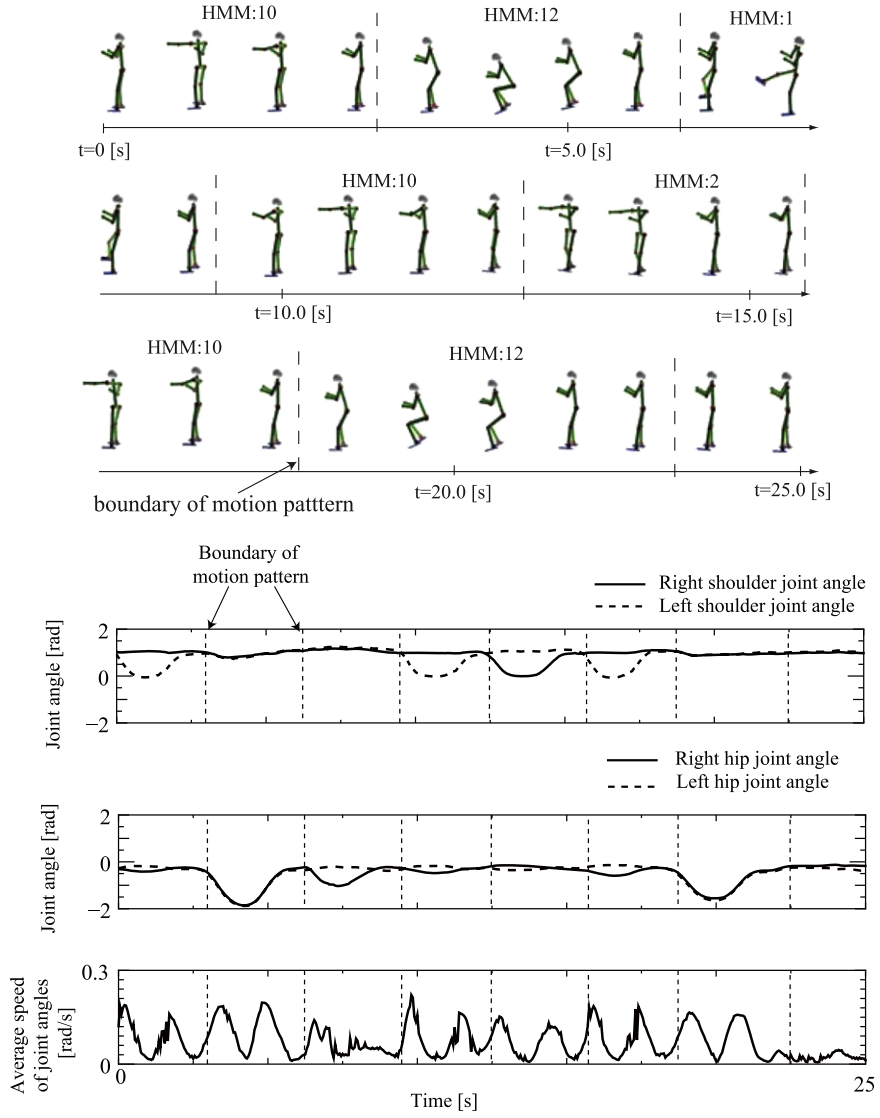


Fig. 4. Snapshots of a human figure show the segmentation results for a sequence of captured motion data. Dashed lines denote the boundaries of motion patterns determined by the proposed segmentation method. The capture data are divided into "left punch", "bend", "right kick", "left punch", "right punch", "left punch" and "bend" motion patterns in real-time. These segmented motion patterns are recorded as the 10th, 12th, 1st, 10th, 2nd, 10th and 12th HMMs respectively. Three graphs at the bottom show the time profiles of the "right shoulder joint angle", "left shoulder joint angle", "right hip joint angle" and "left hip joint angle", as well as the average angular speed of all 20 joints.

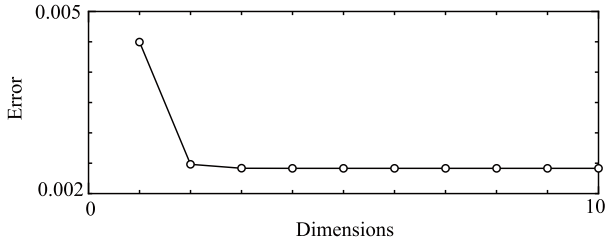


Fig. 5. The relationship between the number of spatial dimensions and distance error.

as described in the next section, and each motion primitive is categorized into the motion symbol with the largest likelihood of generating that motion primitive. The motion labels, such as "right kick", "left kick", "right punch", "left punch", "retract right leg", "retract left leg", and "bend", are manually assigned to the motion symbols. The next section describes the automatic acquisition of the motion symbols in detail. Fig. 6 shows that motion primitives categorized into the same motion group are located close to each other and that motion primitives form a cluster structure of motion patterns.

6. Symbolization of motion through segmentation

The validity of RUS followed by AUS was experimentally verified. A humanoid robot autonomously acquired motion symbols from motion primitives determined as in the previous subsection. We had 20 motion symbols whose parameters were initially set to random values. The competitive learning algorithm incrementally chose one HMM corresponding to a derived motion primitive and optimized the HMM from that motion primitive. In this incremental optimization phase, five HMMs were not optimized at all because these HMMs were not chosen as a motion symbol corresponding to a motion primitive. Therefore, these five HMMs were removed from the set of motion symbols. AUS caused the humanoid robot to acquire 15 motion symbols by using RUS.

We have already constructed a space by locating 333 motion primitives in 4-dimensional space, as described in the previous section. The 15 acquired motion symbols were located in this space

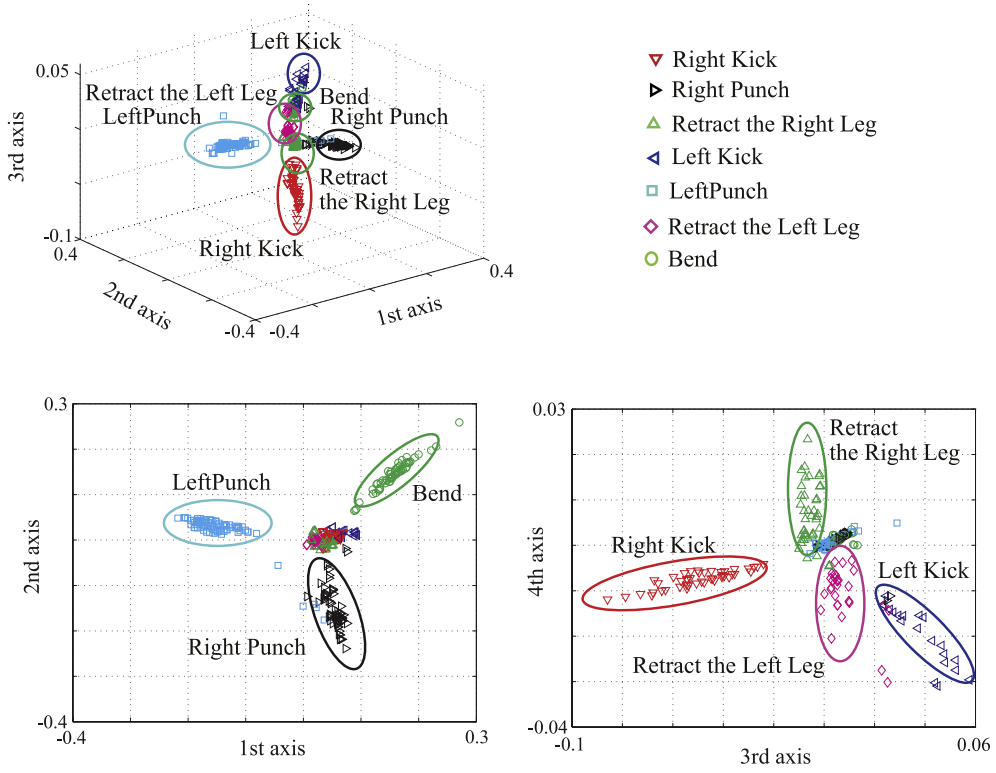


Fig. 6. Space of segmented motion patterns. Segmented motion patterns are converted into HMMs, which are optimized by using one segmented motion pattern as training data. The HMMs are located in 4-dimensional space based on dissimilarities among themselves. The shape and color of each mark indicate the cluster to which each segmented motion pattern belongs. The upper figure projects the 4-dimensional space onto 3-dimensional space for visualization. The lower figures project the 4-dimensional space onto 2-dimensional spaces. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

based on the distance between the two HMMs that represented the motion symbol and motion primitives. Fig. 7 shows the constructed space by locating the motion symbols and the motion primitives in 4-dimensional space; this space is referred to as "motion symbol space". A motion symbol is located within a cluster formed by motion primitives. This associates the HMM developed to a motion symbol that is representative of the motion primitives. The seven motion labels, such as "right kick", "left kick", "right punch", "left punch", "retract right leg", "retract left leg", and "bend" can be manually assigned to the motion symbols. Fifteen motion symbols were acquired. Note that the number of motion labels differs from the number of motion symbols. As shown in Fig. 7, a cluster corresponding to a motion label may include multiple motion symbols. For example, three motion symbols are located in one cluster for the motion label "bend".

We tested motion recognition based on the acquired motion symbols. Fig. 4 shows the experimental result, where seamless human behaviors are segmented into motion primitives and each motion primitive is identified by the motion symbol that has the largest likelihood of containing the motion primitive. Three obtained motion primitives of "left punch" were classified into the 10th motion symbol, and two obtained motion primitives of "bend" were classified into the 12th motion symbol. Similar motion primitives were classified into the same motion symbol.

We also tested motion generation based on the acquired motion symbols. Fig. 8 shows the motions generated by the motion symbols. We selected one motion symbol out of the several motion symbols included in a cluster corresponding to each motion label. The motions of "right kick", "left kick", "right punch", "left punch", "retract right leg", "retract left leg", and "bend" can be generated from these motion symbols. Fig. 9 shows that a small humanoid robot can perform human-like whole-body motion according to the generated motions. This experiment validates the generation of a humanoid robot's whole-body motion from the acquired motion symbols. Therefore, we can confirm that RUS can be used for AUS for a humanoid robot.

7. Segmentation and symbolization of a large motion dataset

A humanoid robot integrated into daily life is expected to memorize many motion symbols so that the robot can recognize and generate a large variety of motions. We tested RUS and AUS on a large motion dataset. We recorded motions performed by three subjects. The dataset consists of 2 h and 48 min of motion data. This motion data was repeatedly given to RUS and AUS in the same manner that in the previous section. We set the numbers of feature HMMs and motion symbols to 300 ($N_f = 300$) and 50 ($N_s = 50$), respectively.

Fig. 12 shows the motion symbol space, where the motion symbols are located in a 6-dimensional space. In Fig. 12, the motion symbols are classified into six groups; figures on each row display only those motion symbols classified into the corresponding group. Additionally, Fig. 10 shows the relation between the number of dimensions of the motion symbol space and the distance error, which is measured by the actual distances among the motion symbols and the distances among locations of the motion symbols in the space. The rate of change of the error ($\gamma = \frac{E_d - E_{d+1}}{E_d}$) converges to $\gamma = 0.003$ in 6-dimensional space. Group 1 in Fig. 11 shows that the motion symbols to which the label "run" can be assigned (λ_{13} and λ_{15}) are located close to the motion symbol to which the label "walk" can be assigned (λ_{16}). Group 4 shows that the motion symbols to which the label "swing a bat" can be assigned (λ_{11}, λ_{14}, and λ_{25}) and the motion symbols to which the label "raise a hand" can be assigned (λ_{10}, λ_{28}, λ_{29}, and λ_{41}) are included in the same group. Group 5 includes motion symbols representing "sitting" (λ_3, λ_5, and λ_7). Group 6 includes the motion symbols representing "stretching exercise" (λ_4, λ_{36}, and λ_{38}). Motion symbols

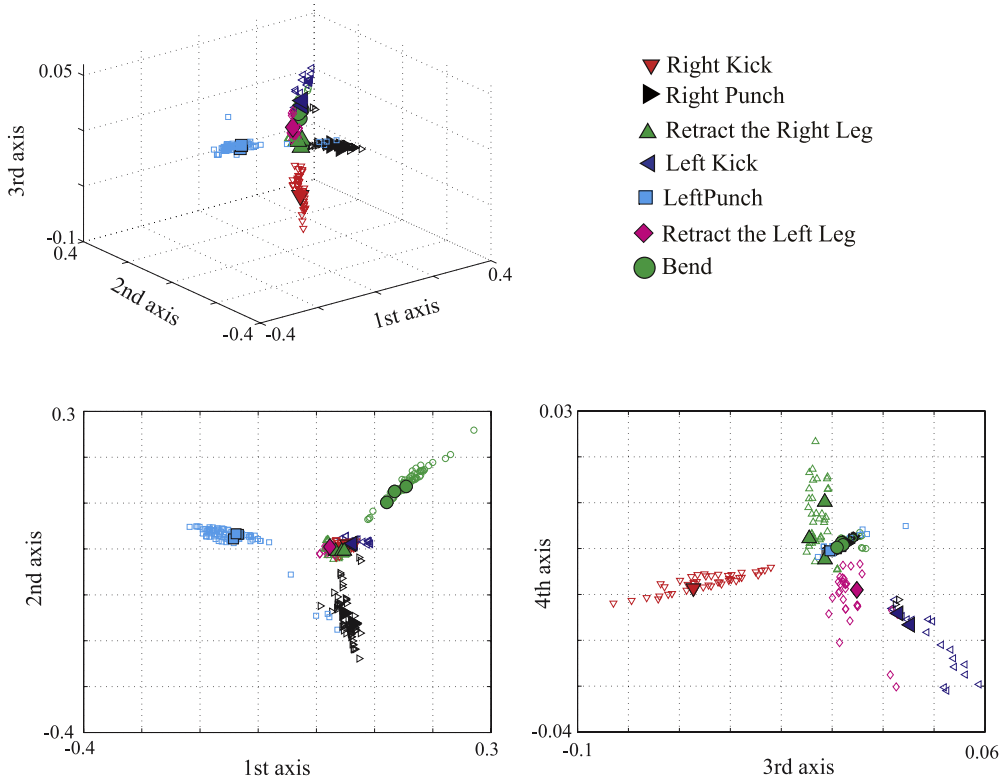


Fig. 7. Motion symbol space consists of motion symbols and segmented motion primitives. Filled marks and blank marks correspond to motion symbols and motion primitives, respectively. Motion symbols are located close to the cluster area containing the motion primitives that are recognized as motion symbols.

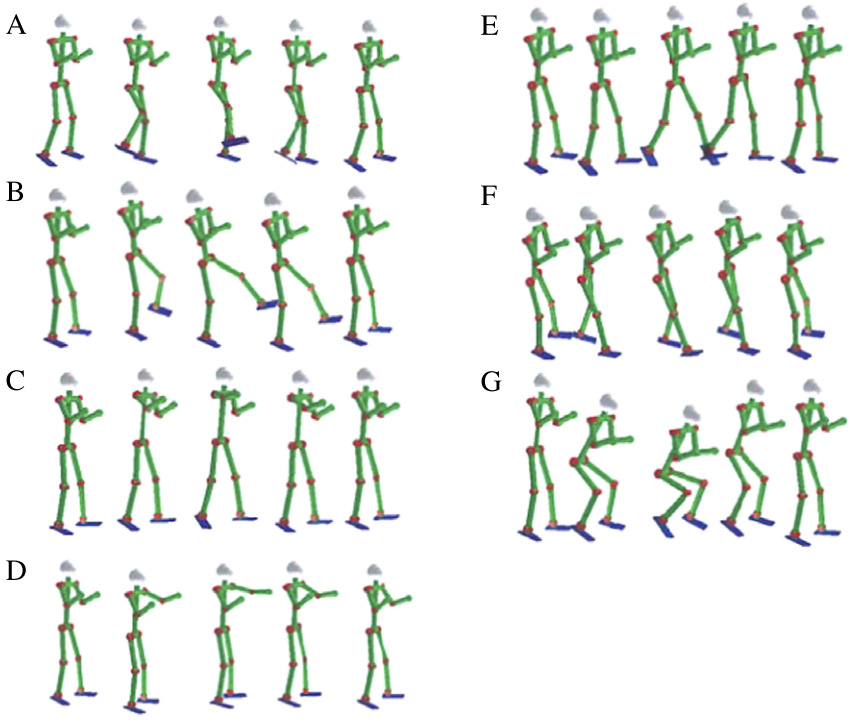


Fig. 8. Generated motion primitive for each motion symbol. The motion primitives of (A)–(G) can be subjectively classified as "right kick", "left kick", "right punch", "left punch", "retract the right leg", "retract the left leg", and "bend".

with the same label are located close to each other in the motion symbol space.

The motion primitives determined by RUS are also located in the motion symbol space. The location occurs by conversion of the motion primitives to HMMs, and measurement of distances between the motion primitives and the motion symbols. Motion primitives that are recognized as the same motion symbol form clusters in the motion symbol space. The degree of separation $S = \frac{\sigma_{intra}}{\sigma_{inner}}$ can be computed by inner-class variance σ_{inner} and inter-class variance σ_{intra}. The large degree of separation implies that motion

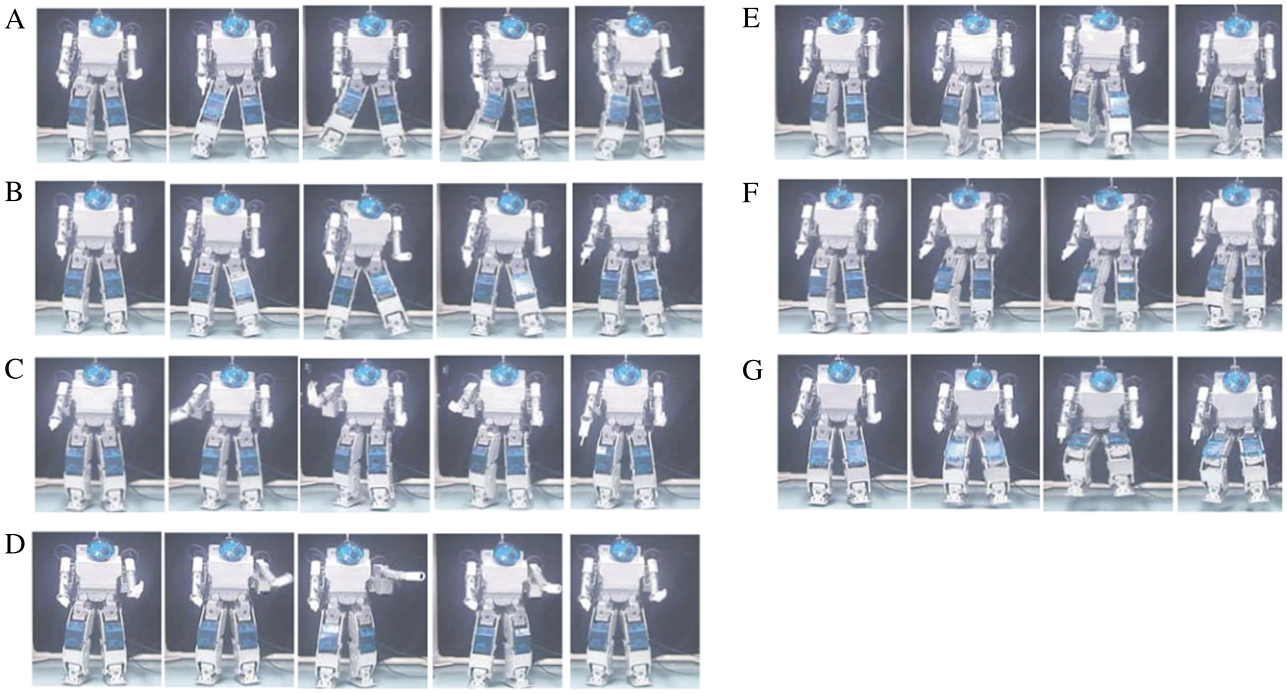


Fig. 9. The humanoid robot can perform each motion primitive generated by the motion symbol. (A)–(G) depict the generated motion patterns of "right kick", "left kick", "right punch", "left punch", "retract right leg", "retract left leg", and "bend", respectively.

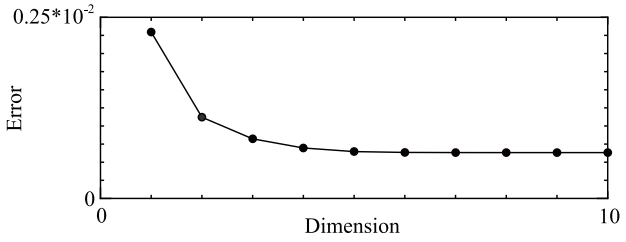


Fig. 10. Relationship between the number of spatial dimensions and distance error.

primitives identified as the same motion symbol are close to each other and that motion primitives identified as different motion symbols are located distantly from each other. The large degree of separation verifies that seamless human whole-body motion can be segmented to a sequence of motion primitives, which are frequently observed motion patterns.

Fig. 13 shows two degrees of separation for motion primitives determined by RUS, and for motion primitives determined by a random segmentation method. Note that the random segmentation method is designed such that the mean and variance of the length of the motion primitives derived by the random segmentation method is equal to the mean and variance of the lengths of the primitives determined by RUS. In this evaluation, we set the number of motion symbols to 10, 20, 30, 40, and 50, and then computed separation measures for each case. Fig. 13 demonstrates that the degree of separation for RUS is larger than that for random segmentation in all conditions.

8. Effects of parameters of RUS and symbolization

8.1. The number of nodes in a motion symbol

The performance of motion recognition and generation from motion symbols depends on the number of nodes in the HMMs for the motion symbols. The relationship between the likelihood of training motion data being generated by the corresponding HMM and the number of nodes is shown by Fig. 14. For this figure, "bend" motion data, 300 frames captured at 100 (fps), was used. As the number of nodes increases, the likelihood becomes larger. The likelihood converges at five nodes. Based on this, we choose 10 as a sufficient number of nodes for each motion symbols.

8.2. Sampling rate

We tested the influence of the data sampling rate on the performance of RUS and AUS. We used an optical motion capture system with a sampling rate of 10 (ms) to measure a human subject performing the "bending" motion. The measured original data was down-sampled to motion data with sampling times of 20 (ms), 30 (ms), $\cdots$, 300 (ms). The motion data were given to the corresponding HMMs as training data. We compared the original motion data with motion data generated by the HMMs. The comparison looks for correspondence between motion frames in the original motion data and in the generated motion data because the sampling rate of the original motion data is different from that of the HMMs. The HMM λ^{χ} training motion data with sampling time of χ (ms) generates motion data with the same sampling time; then, the expected time to stay on the ith node, τ_i^{χ}, can be calculated using the probability of transition from the ith node to the ith node, a_{ii}^{χ}, as follows:

$$\tau_i^{\chi} = \frac{1}{1 - a_{ii}^{\chi}}. \tag{16}$$

To adapt the HMM λ^{χ} to motion data with a sampling time of 10 (ms), the transition probability for the expected time to stay on the ith node, τ_i^{χ}, is multiplied by $\frac{\chi}{10}$. The result of this is that the original motion data and the motion data generated by the modified HMM are of the same length. This makes it possible to calculate the correspondence between two frames of the original motion data and the generated motion data. Fig. 15 shows average errors between corresponding frames and sampling times. For sampling times in the range 10 (ms) to 300 (ms), the average error remains below 0.05 (rad). A sampling time of 50 (ms) has sufficient temporal resolution for human whole-body motions.

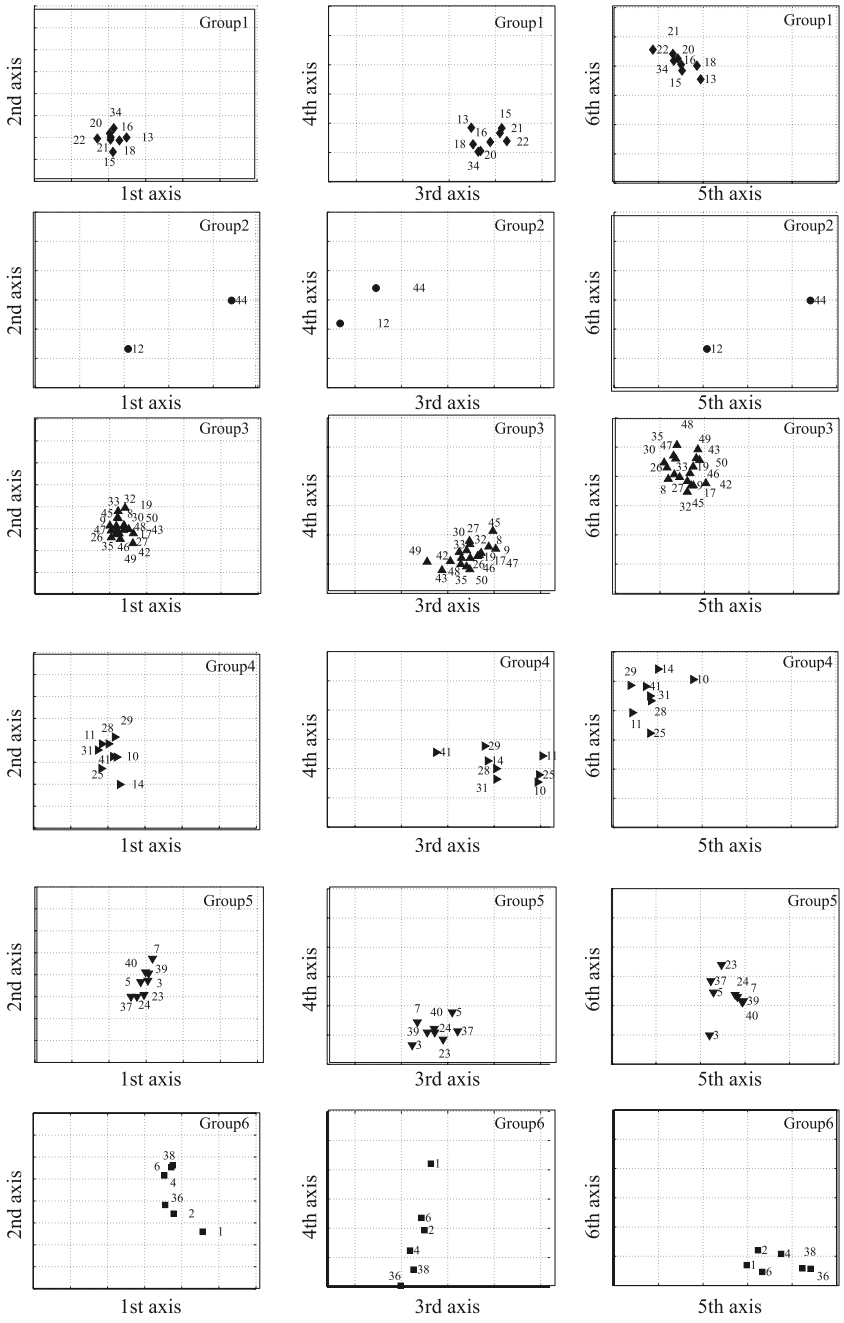


Fig. 11. 50 motion symbols are located in 6-dimensional space. The motion symbols are classified into 6 groups. Each figure shows the symbol space with the motion symbols in the same group.

8.3. Number of feature HMMs

The performance of RUS depends on the number of feature HMMs. We investigated the relationship between RUS performance and the number of the feature HMMs. We introduce two measures, "accuracy" and "completeness", to quantify the performance; these are analogous to those Brent et al. adopted to segment English sentences [25]. English sentences have clear boundaries for words, and it is easy to compare the detected boundaries with the correct boundaries. However, the boundaries of motion primitives in seamless human whole-body motion are unclear. A human subject was asked to identify the boundaries of the motion primitives, and detected boundaries within 1.0 (s) of these chosen boundaries are regarded as correctly detected. Comparison between the automatically detected boundaries and the manually identified boundaries yields the accuracy and completeness. For detection by RUS, True positive, false positive, and false negative are defined as a correctly detected boundary, an incorrectly detected boundary, and an undetected boundary, respectively. From the counts of the true positives (N_{tp}), false positives (N_{fp}), and false negatives (N_{fn}), the accuracy and completeness are defined as follows:

$$Z_{accuracy} = \frac{N_{tp}}{N_{tp} + N_{fp}} \tag{17}$$

$$Z_{completeness} = \frac{N_{tp}}{N_{tp} + N_{fn}}. \tag{18}$$

Fig. 16 shows the accuracy and completeness for a 4 min motion capture. We set the number of the feature HMMs to 10, 20, . . . , 110. Ten feature HMMs led to low completeness

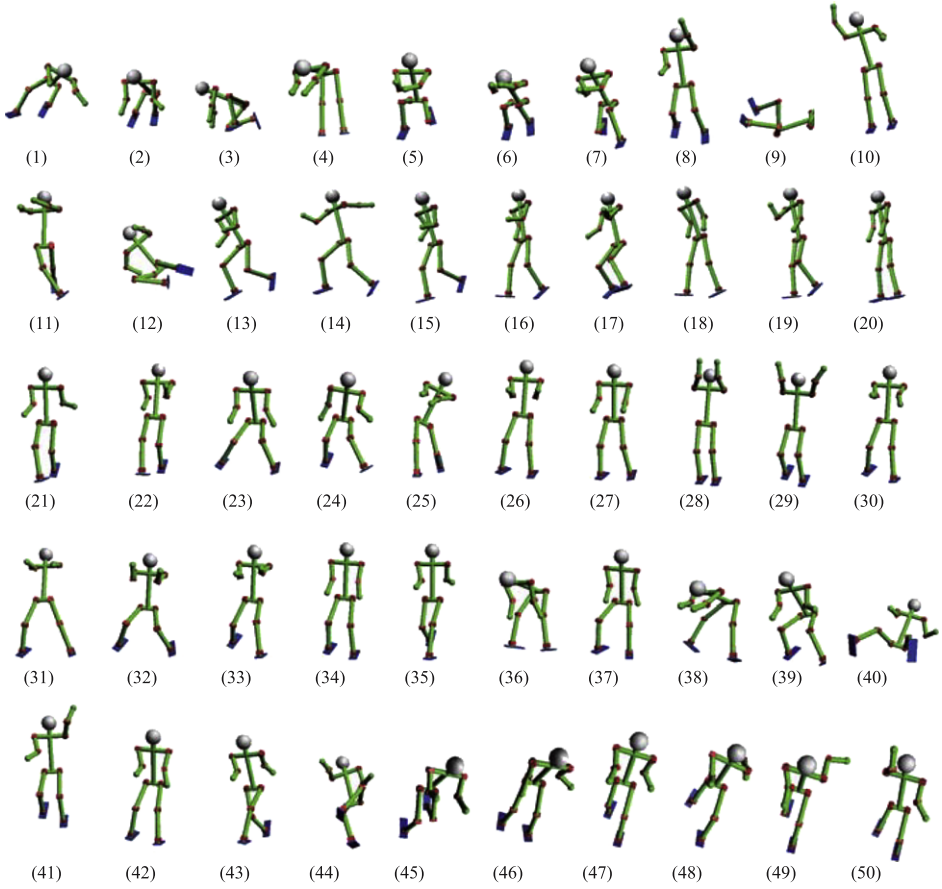


Fig. 12. Each snapshot shows a motion pattern generated by a motion symbol.

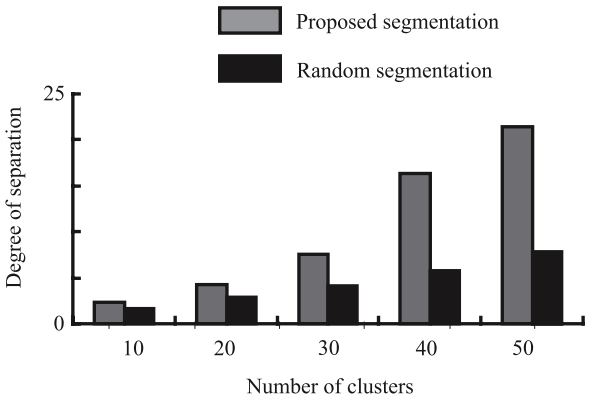


Fig. 13. Comparison between proposed segmentation and random segmentation in terms of degree of separation. The score can be computed as the ratio of intra-class variation to inner-class variation. Higher scores indicate better clustering.

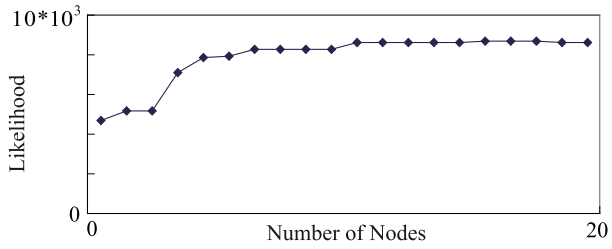


Fig. 14. Relationship between the number of HMM nodes and the likelihood that the HMM generates the human motion data. As the number of the HMM nodes increases up to 5, the likelihood increases.

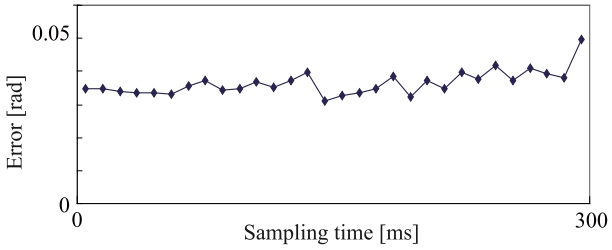


Fig. 15. Relationship between sampling time and average error of joint angles for original training data and data generated by HMM. Average errors remain below 0.05 rad.

because the derived motion primitives are long. As the number of the feature HMMs was increased, completeness was improved. More than 20 HMMs resulted in a high accuracy, 90%, and good completeness, 70%.

8.4. Size of feature vector

A feature vector $\boldsymbol{x}$ is formed by aligning M pieces of binary-valued vectors $\hat{\boldsymbol{x}}$. The variable M represents the time constant of RUS. We conducted the experiment to determine the relationship between the size of the feature vector and the RUS. Fig. 17 shows the size of the feature vector and the corresponding performance of RUS. We chose the degree of separation as the measure of performance. By incrementing the size of the feature vectors from two to ten and comparing RUS to random segmentation, we see that RUS provided motion primitives with a large degree of separation. Additionally, this experiment verified that RUS does not require a special tuning of the size of the feature vector.

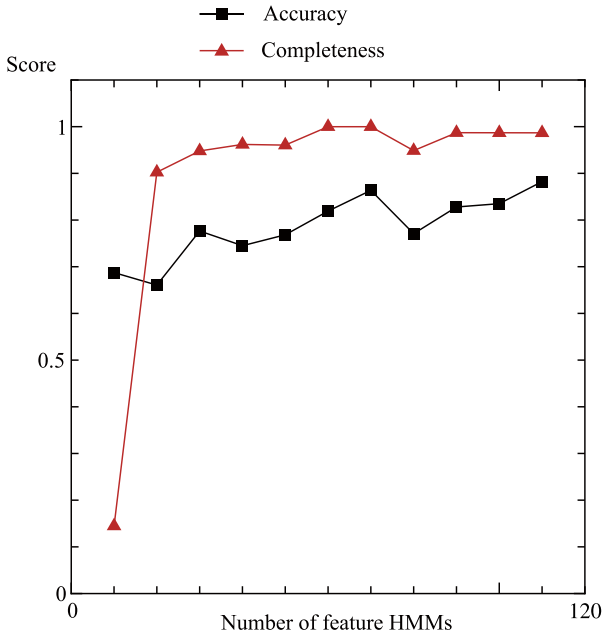


Fig. 16. Evaluation scores of accuracy and completeness. The number of feature HMMs is set from 10 to 110.

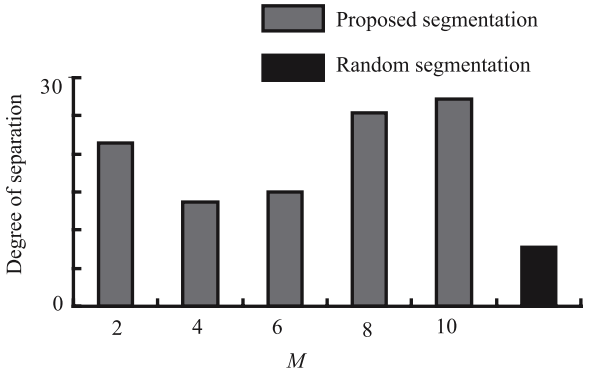


Fig. 17. Correlation relationship between degree of separation and temporal size.

9. Conclusion

The contributions of this study can be summarized as follows.

1. We developed the RUS method for segmenting human whole-body motions. RUS converts seamless human motion into a sequence of feature HMMs, extracts causality between the feature HMMs, and predicts the motion following an observed movement by using the derived causality. A motion with a large prediction uncertainty is detected as the boundary of a motion primitive. Experiments verified that, compared with random segmentation, RUS has better segmentation performance in terms of clustering. Motion primitives determined by RUS form a cluster structure of motions with a large degree of separation. The experiments also demonstrated that boundaries detected by RUS coincide more closely with manually selected boundaries than random segmentation.
2. We applied RUS to AUS of motion patterns. The derived motion primitives are classified as one of the HMMs, referred to as motion symbols; the parameters of HMMs are incrementally optimized from the motion primitives by competitive learning. The HMMs make it possible to recognize observed motion as a motion symbol and to generate human-like motion. The experiments demonstrated that motion symbols can abstract motion primitives located close to each other in the motion symbol space, that similar motions are recognized as the same motion symbol, and that motion symbols can generate human-like whole-body motions for a humanoid robot.
3. We conducted experiments to explore the relationship between variables of a framework integrating RUS and AUS and the performance of segmentation and symbolization. We chose, as our variables of interest, the number of nodes in an HMM representing a motion symbol, the sampling rate when capturing human whole-body motions, the number of feature HMMs, and the size of a feature vector; we investigated the effects of these variables on the performance of the framework. This discussion provides a guide for designing the controller for an intelligent humanoid robot by using our proposed framework.

Our proposed framework has a variety of parameters to be tuned in order to derive usable RUS and AUS. We tested several parameters such as the sampling rate to capture human motions, the dimensionality of the feature vector, the number of feature HMMs, the number of nodes in each motion symbol HMM, for the segmentation and autonomous symbolization of human whole body motions. The experiments demonstrated that the high capturing rate, the high dimensionality of the feature vector, the large number of feature HMMs and the node in the motion symbol HMMs are likely to lead to the modest performance of the segmentation and symbolization. However, these parameter setting consume more time, and we need to find the reasonable parameters taking into consideration both performance and computational cost. Additionally we have not tested the dependence of performance on each of all the parameters yet. We need to investigate another parameters, and tunes them for the specific domain of human actions.

Acknowledgment

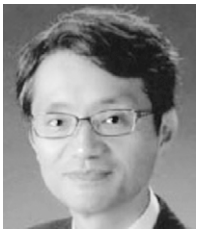
This research was partially supported by a Grant-in-Aid for Young Scientists (A) (No. 26700021) from the Japan Society for the Promotion of Science, and by the Strategic Information and Communications R&D Promotion Program (No. 142103011) of the Ministry of Internal Affairs and Communications.

References

[1] Y. Kuniyoshi, M. Inaba, H. Inoue, Learning by watching: Extracting reusable task knowledge from visual observation of human performance, IEEE Trans. Robot. Autom. 10 (6) (1994) 799–822.
[2] J. Morimoto, K. Doya, Hierarchical reinforcement learning for motion learning: learning "stand-up" trajectories, Adv. Robot. 13 (3) (1999) 267–268.
[3] M.J. Mataric, Getting humaniods to move and imitate, IEEE Intell. Syst. 15 (4) (2000) 18–24.
[4] M. Haruno, D. Wolpert, M. Kawato, MOSAIC model for sensorimotor learning and control, Neural Comput. 13 (2001) 2201–2220.
[5] J. Tani, M. Ito, Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment, IEEE Trans. Syst. Man Cybern. A 33 (4) (2003) 481–488.
[6] T. Inamura, I. Toshima, H. Tanie, Y. Nakamura, Embodied symbol emergence based on mimesis theory, Int. J. Robot. Res. 23 (4) (2004) 363–377.
[7] A. Billard, S. Calinon, F. Guenter, Discriminative and adaptive imitation in uni-manual and bi-manual tasks, Robot. Auton. Syst. 54 (2006) 370–384.
[8] S. Chiappa, J. Kober, J. Peters, Using Bayesian dynamical systems for motion template libraries, in: Advances in Neural Information Processing Systems, Vol. 21, 2008, pp. 297–304.
[9] V. Gallese, A. Goldman, Mirror neuron and the simulation theory of mind-reading, Trends Cogn. Sci. 2 (12) (1998) 493–501.
[10] G. Rizzolatti, L. Fogassi, V. Gallese, Neurophysiological mechanisms underlying the understanding and imitation of action, Nat. Rev. (2001) 661–670.
[11] M. Donald, Origin of the Modern Mind, Harvard University Press, Cambridge, 1991.
[12] T. Mori, Y. Segawa, M. Shimosaka, T. Sato, Segmentation of sequential daily-actions based on hidden Markov models and human body hierarchy, in: Proceedings of The Second International Workshop on Man–Machine Symbiotic Systems, 2004, pp. 207–218.

[13] J. Kohlmorgen, S. Lemm, A dynamic HMM for On-line segmentation of sequential data, in: Advances in Neural Information Processing System, Vol. 14, 2001.
[14] D. Kulic, W. Takano, Y. Nakamura, On-line segmentation and clustering from continuous observation of whole body motions, IEEE Trans. Robot. 25 (5) (2009) 1158–1166.
[15] B. Janus, Y. Nakamura, Unsupervised probabilistic segmentation of motion data for mimesis modeling, in: Proceedings of IEEE International Conference on Advanced Robotics, 2005, pp. 411–417.
[16] K. Grave, S. Behnke, Incremental action recognition and generalizing motion generation based on goal-directed features, in: Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 751–757.
[17] T.S. Wang, H.Y. Shum, Y.Q. Xu, N.N. Zheng, Unsupervised analysys of human gestures, in: Proceedings of the Second IEEE Pacific-Rim Conference on Multimedia, Vol. 10, 2011, pp. 174–181.
[18] T. Kim, S.I. Park, S.Y. Shin, Nonmetric individual differences multidimensional scaling: An alternating least squres method with optimal scaling features, ACM Trans. Graph. 22 (3) (2003) 392–401.
[19] T. Shiratori, A. Nakazawa, K. Ikeuchi, Detecting dance motion structure through music analysisg, in: Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Vol. 5, 2004, pp. 857–862.
[20] M.R. Brent, Speech segmentation and word discovery: a computational perspective, Trends Cogn. Sci. 3 (8) (1999) 294–301.
[21] M. Redington, N. Charter, Probabilistic and distributional approaches to language acquisition, Trends Cogn. Sci. 1 (7) (1997) 273–281.
[22] M.H. Christiansen, S. Curtin, Transfer of learning: rule acquisition or statistical learning, Trends Cogn. Sci. 3 (8) (1999) 289–290.
[23] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (1990) 179–211.
[24] P. Cairns, R. Shillcock, N. Chater, J. Levy, Bootstrapping word boundaries: A bottom-up corpus-based approach to speech segmentation, Cogn. Psychol. 33 (2) (1997) 111–153.
[25] M.R. Brent, An efficient, probabilistically sound algorithm for segmentation and word discovery, Mach. Learn. 34 (1999) 71–106.
[26] L. Rabiner, B.H. Juang, Fundamentals of Speech Recognition, Prentice Hall Signal Processing Series, 1993.
[27] S. Calinon, A. Billard, Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM, in: Proceedings of the 22th International Conference on Machine Learning, 2005, pp. 105–112.
[28] K. Yamane, Y. Nakamura, Natural motion animation through constraining and deconstraining at Will, IEEE Trans. Vis. Comput. Graphics 9 (3) (2003) 352–360.

Wataru Takano is an Assistant Professor at Department of Mechano-Informatics, School of Information Science and Technology, University of Tokyo. He was born in Kyoto, Japan, in 1976. He received the B.S and M.S degrees from Kyoto University, Japan, in precision engineering in 1999 and 2001, Ph.D.degree from Mechano-Informatics, the University of Tokyo, Japan, in 2006. He was a Project Assistant Professor at the University of Tokyo from 2006 to 2007, and a Researcher on Project of Information Environment and Humans, Presto, Japan Science and Technology Agency from 2010. His fields of research include kinematics, dynamics, artificial intelligence of humanoid robots, and intelligent vehicles. He is a member of IEEE, Robotics Society of Japan, and Information Processing Society of Japan. He has been the chair of Technical Committee of Robot Learning, IEEE RAS.



Yoshihiko Nakamura is a Professor at Department of Mechano-Informatics, School of Information Science and Technology, University of Tokyo. He was born in Osaka, Japan, in 1954. He received the B.S., M.S., and Ph.D. degrees from Kyoto University, Japan, in precision engineering in 1977, 1978, and 1985, respectively. He was an Assistant Professor at the Automation Research Laboratory, Kyoto University, from 1982 to 1987. He joined the Department of Mechanical and Environmental Engineering, University of California, Santa Barbara, in 1987 as an Assistant Professor, and became an Associate Professor in 1990. He was also a co-director of the Center for Robotic Systems and Manufacturing at UCSB. He moved to University of Tokyo as an Associate Professor of Department of Mechano-Informatics, University of Tokyo, Japan, in 1991. His fields of research include the kinematics, dynamics, control and intelligence of robots—particularly, robots with non-holonomic constraints, computational brain information processing, humanoid robots, human-figure kinetics, and surgical robots. He is a member of IEEE, ASME, SICE, Robotics Society of Japan, the Institute of Systems, Control, and Information Engineers, and the Japan Society of Computer Aided Surgery. He was honored with a fellowship from the Japan Society of Mechanical Engineers. Since 2005, he has been the president of Japan IFToMM Congress. He is a foreign member of the Academy of Engineering in Serbia and Montenegro.