frontiers

Segmenting Continuous Motions with Hidden Semi-Markov Models and Gaussian Processes

Tomoaki Nakamura^{1,*}**, Takayuki Nagai**^{1}**, Daichi Mochihashi**^{2}**, Ichiro Kobayashi**^{3}**, Hideki Asoh**^{4}**, and Masahide Kaneko**^{1}

[1] *Department of Mechanical Engineering and Intelligent Systems, The University of Electro-Communications, Chofu-shi, Japan*
[2] *Department of Mathematical Analysis and Statistical Inference, Institute of Statistical Mathematics, Tachikawa, Japan*
[3] *Department of Information Sciences, Faculty of Sciences, Ochanomizu University, Bunkyo-ku, Japan*
[4] *Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan*

Correspondence*:
Tomoaki Nakamura
tnakmaura@uec.ac.jp

ABSTRACT

Humans divide perceived continuous information into segments to facilitate recognition. For example, humans can segment speech waves into recognizable morphemes. Analogously, continuous motions are segmented into recognizable unit actions. People can divide continuous information into segments without using explicit segment points. This capacity for unsupervised segmentation is also useful for robots, because it enables them to flexibly learn languages, gestures, and actions. In this paper, we propose a Gaussian process-hidden semi-Markov model (GP-HSMM) that can divide continuous time series data into segments in an unsupervised manner. Our proposed method consists of a generative model based on the hidden semi-Markov model (HSMM), the emission distributions of which are Gaussian processes (GPs). Continuous time series data is generated by connecting segments generated by the GP. Segmentation can be achieved by using forward filtering-backward sampling to estimate the model's parameters, including the lengths and classes of the segments. In an experiment using the CMU motion capture dataset, we tested GP-HSMM with motion capture data containing simple exercise motions; the results of this experiment showed that the proposed GP-HSMM was comparable with other methods. We also conducted an experiment using karate motion capture data, which is more complex than exercise motion capture data; in this experiment, the segmentation accuracy of GP-HSMM was 0.92, which outperformed other methods.

Keywords: Motion segmentation, Gaussian process, hidden semi-Markov model, motion capture data

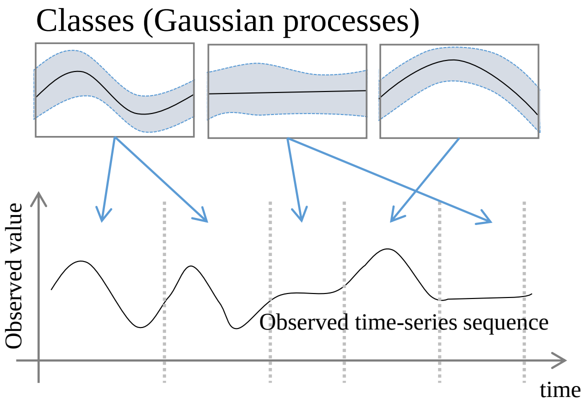


Figure 1. Overview of the proposed GP-HSMM.

1 INTRODUCTION

Human beings typically divide perceived continuous information into segments to enable recognition. For example, humans can segment speech waves into recognizable morphemes. Similarly, continuous motions are segmented into recognizable unit actions. In particular, motions are divided into smaller components called motion primitives, which are used for imitation learning and motion generation (**??**). It is possible for us to divide continuous information into segments without using explicit segment points. This capacity for unsupervised segmentation is also useful for robots, because it enables them to flexibly learn languages, gestures, and actions.

However, segmentation of time series data is a difficult task. When time series data is segmented, the data points in the sequence must be classified, and each segment's start and end points must be determined. Moreover, each segment affects other segments because of the nature of time series data. Hence, segmentation of time series data requires the exploration of all possible segment lengths and classes. However, this exploration process is difficult; in many studies, the lengths are not estimated explicitly or heuristics are used to reduce computational complexity. Furthermore, in the case of motions, the sequences vary because of dynamic characteristics, even though the same movements are performed. For segmentation of actual human motions, we must address such variations.

In this paper, we propose GP-HSMM (Gaussian process - hidden semi-Markov model), a novel method to divide time series motion data into unit actions by using a stochastic model to estimate their lengths and classes. The proposed method involves a hidden semi-Markov model (HSMM) with a Gaussian process (GP) emission distribution, where each state represents a unit action. Fig. **??** shows an overview of the proposed GP-HSMM. The observed time series data is generated by connecting segments generated by each class. The segment points and segment classes are estimated by learning the parameters of the model in an unsupervised manner. Forward filtering-backward sampling (**?**) is used for the learning process; the segment lengths and segment classes are determined by sampling them simultaneously.

2 RELATED WORK

Various studies have focused on learning motion primitives from manually segmented motions (**??**). Manschitz et al. proposed a method to generate sequential skills by using motion primitives that are learned in a supervised manner. Gräve et al. proposed segmenting motions using motion primitives that

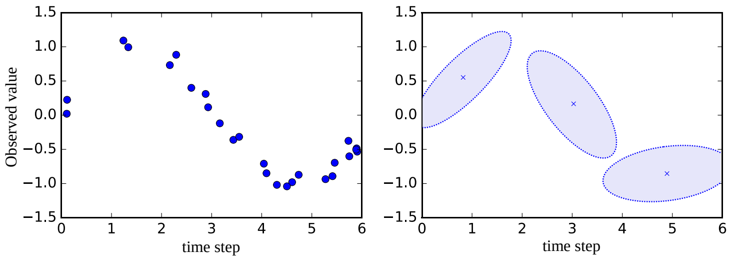


Figure 2. Example of representation of time series data by HMM. Left: Data points for learning HMM. Right: Mean and standard deviation learned by HMM.

are learned by a supervised hidden Markov model. In these studies, the motions are segmented and labeled in advance. However, we consider that it is difficult to segment and label all possible motion primitives.

Additionally, some studies have proposed unsupervised motion segmentation. However, these studies rely on heuristics. For instance, Wächter et al. have proposed a method to segment human manipulation motions based on contact relations between the end-effectors and objects in a scene (**?**); in their method, the points at which the end-effectors make contact with an object are determined as boundaries of motions. We believe this method works well in limited scenes; however, there are many motions, such as gestures and dances, in which objects are not manipulated. Lioutikov et al. proposed unsupervised segmentation; however, to reduce computational costs, this technique requires the possible boundary candidates between motion primitives to be specified in advance (**?**). Therefore, the segmentation depends on those candidates, and motions cannot be segmented correctly if the correct candidates are not selected. In contrast, our proposed method does not require such candidates; all possible cutting points are considered by use of forward filtering-backward sampling, which uses the principles of dynamic programming. In some methods (**????**), motion features (such as the zero velocity of joint angles) are used for motion segmentation. However, these features cannot be applied to all motions. Takano et al. use the error between actual movements and predicted movements as the criteria for specifying boundaries (**?**). However, the threshold must be manually tuned according to the motions to be segmented. Moreover, they used an HMM that is a stochastic model. We consider such an assumption to be unnatural from the viewpoint of stochastic models, and boundaries should be determined based on a stochastic model. In our proposed method, we do not use such heuristics and assumptions, and instead formulate the segmentation based on a stochastic model.

Fox et al. have proposed unsupervised segmentation for the discovery of a set of latent, shared dynamical behaviors in multiple time series data (**?**). They introduce a beta process, which represents a share of motion primitives in multiple motions, into autoregressive HMM. They formulate the segmentation using a stochastic model, and no heuristics are used in their proposed model. However, in their proposed method, continuous data points that are classified into the same states are extracted as segments, and the lengths of the segments are not estimated. The states can be changed in the short term, and therefore shorter segments are estimated. They reported that some true segments were split into two or more categories, and that those shorter segments were bridged in their experiment. On the other hand, our proposed method classifies data points into states, and uses HSMM to estimate segment lengths. Hence, our proposed method can prevent states from being changed in the short term.

Matsubara et al. proposed an unsupervised segmentation method called AutoPlait (**?**). This method uses multiple HMMs, each of which represents a fixed pattern; moreover, transitions between the HMMs are allowed. Therefore, time series data is segmented at points at which the state is changed to another HMM's state. However, we believe that HMMs are too simple to represent complicated sequences such as motions. Fig. ?? illustrates an example of representation of time series data by HMM. The graph on the right in Fig. ?? represents the mean and standard deviation learned by HMM from data points shown in the graph on the left. HMM represents time series data using only the mean and standard deviation; therefore, details of time series data can be lost. Therefore, we use Gaussian processes, which are non-parametric methods that can represent complex time series data.

The field of natural language processing has also produced literature related to sequence data segmentation. For example, unsupervised morphological analysis has been proposed for segmenting sequence data (???). Goldwater et al. proposed a method to divide sentences into words by estimating the parameters of a 2-gram language model based on a hierarchical Dirichlet process. The parameters are estimated in an unsupervised manner by Gibbs sampling (?). Mochihashi et al. proposed a nested Pitman-Yor language model (NPYLM) (?). In this method, parameters of an n-gram language model based on the hierarchical Pitman-Yor process are estimated via the forward filtering-backward sampling algorithm. NPYLM can thus divide sentences into words more quickly and accurately than the method proposed in (?). Moreover, Uchiumi et al. extended the NPYLM to a Pitman-Yor hidden semi-Markov model (PY-HSMM) (?) that can divide sentences into words and estimate the parts of speech (POS) of the words by sampling not only words, but also POS in the sampling phase of the forward filtering-backward sampling algorithm. However, these relevant studies aimed to divide symbolized sequences (such as sentences) into segments, and did not consider analogous divisions in continuous sequence data, such as that obtained by analyzing human motion.

Taniguchi et al. proposed a method to divide continuous sequences into segments by utilizing NPYLM (?). In their method, continuous sequences are discretized and converted into discrete-valued sequences using the infinite hidden Markov model (?). The discrete-valued sequences are then divided into segments by using NPYLM. In this method, motions can be recognized by the learned model, but cannot be generated naively because they are discretized. Moreover, segmentation based on NPYLM does not work well if errors occur in the discretization step.

Therefore, we propose a method to divide a continuous sequence into segments without using discretization. This method divides continuous motions into unit actions. Our proposed method is based on HSMM, the emission distribution of which is GP, which represents continuous unit actions. To learn the model parameters, we use forward filtering-backward sampling, and segment points and classes are sampled simultaneously. However, our proposed method also has limitations. One limitation is that the method requires the number of motion classes to be specified in advance. It is estimated automatically in methods such as (?) and (?). Another limitation is that computational costs are very high, owing to the numerous recursive calculations. We discuss these limitations in the experiments.

3 GAUSSIAN PROCESS-HIDDEN SEMI-MARKOV MODEL

Fig. ?? shows a graphical representation of the proposed GP-HSMM. In this figure, $c_j (j = 1, 2, \cdots, J)$ denotes classes of segments, and each segment $\boldsymbol{x}_j$ is generated by a Gaussian process, with parameters denoted by $\boldsymbol{X}_c$ and given by the following generative process:

$$c_j \sim P(c|c_{j-1}), \tag{1}$$

$$\boldsymbol{x}_j \sim \mathcal{GP}(\boldsymbol{x}|\boldsymbol{X}_{c_j}), \tag{2}$$

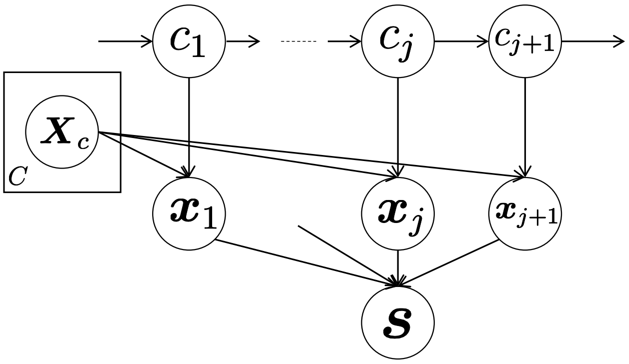


Figure 3. Graphical representation of the proposed GP-HSMM.

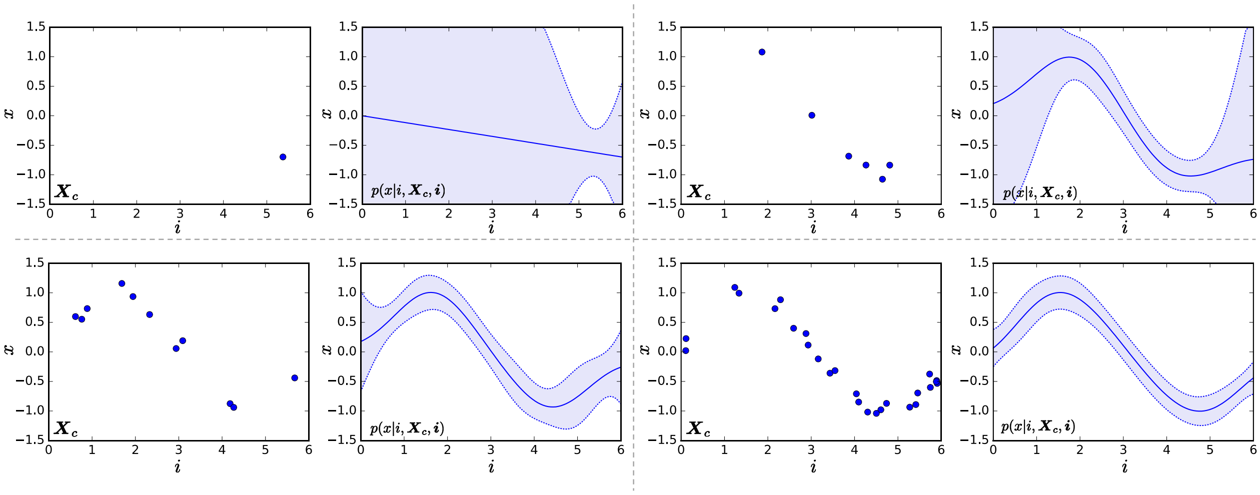


Figure 4. Examples of Gaussian processes. Left graph in each pair of graphs represents learning data points $(\boldsymbol{i}, \boldsymbol{X}_c)$. Right graph shows the learned probabilistic distribution $p(x|i, \boldsymbol{X}_c, \boldsymbol{i})$; the solid line represents the mean, and the blue region represents the range of standard deviation.

where $\boldsymbol{X}_c$ represents a set of segments classified into class c. Segments are generated by this generative process, and the observed time-series data $\boldsymbol{s}$ is generated by connecting the segments.

3.1 Gaussian Process

In this study, we utilize Gaussian process regression, which learns emission x_i of time step i in a segment. This makes it possible to represent each unit action as part of a continuous trajectory. If we obtain pairs $(\boldsymbol{i}, \boldsymbol{X}_c)$ of emissions x_i of time step i of segments belonging to the same class c, a predictive distribution whereby the emission of time step i becomes x follows a Gaussian distribution.

$$p(x|i, \boldsymbol{X}_c, \boldsymbol{i}) \propto \mathcal{N}(\boldsymbol{k}^T \boldsymbol{C}^{-1} \boldsymbol{i}, c - \boldsymbol{k}^T \boldsymbol{C}^{-1} \boldsymbol{k}), \tag{3}$$

where $k(\cdot, \cdot)$ represents the kernel function and $\boldsymbol{C}$ is a matrix whose elements are

$$C(i_p, i_q) = k(i_p, i_q) + \beta^{-1} \delta_{pq}. \tag{4}$$

β is a hyperparameter that represents noise in the observation. In Eq. (??), $\boldsymbol{k}$ is a vector containing the elements $k(i_p, i)$, and c is a scalar value $k(i, i)$. Using the kernel function, GP can learn a time-series sequence that contains complex changes. We use the following Gaussian kernel, which is generally used for Gaussian process regression:

$$k(i_p, i_q) = \theta_0 \exp(-\frac{1}{2}\theta_1 ||i_p - i_q||^2) + \theta_2 + \theta_3 i_p i_q, \tag{5}$$

where θ_* represents parameters of the kernel. Fig. ?? shows examples of Gaussian processes. The left graph in each pair of graphs represents learning data points $(\boldsymbol{i}, \boldsymbol{X}_c)$, and the right graph shows the learned probabilistic distribution $p(x|i, \boldsymbol{X}_c, \boldsymbol{i})$. One can see that the standard deviation decreases with an increase in the number of learning data points. If the emission of time step i is multidimensional vector $\boldsymbol{x} = (x_0, x_1, \cdots)$, we assume that each dimension is generated independently, and a predictive distribution $\mathcal{GP}(\boldsymbol{x}|\boldsymbol{X}_c)$ is computed as follows:

$$\begin{aligned} \mathcal{GP}(\boldsymbol{x}|\boldsymbol{X}_c) &= p(x_0|i, \boldsymbol{X}_{c,0}, \boldsymbol{i}_c) \\ &\quad \times p(x_1|i, \boldsymbol{X}_{c,1}, \boldsymbol{i}_c) \\ &\quad \times p(x_2|i, \boldsymbol{X}_{c,2}, \boldsymbol{i}_c) \cdots . \end{aligned} \tag{6}$$

Based on this probability, similar segments can be classified into the same class.

3.2 Learning of GP-HSMM

3.2.1 Blocked Gibbs Sampler

Segments and classes of segments in the observed sequences are estimated based on dynamic programming and sampling. For efficient sampling, we use the blocked Gibbs sampler, which samples segments and their classes in an observed sequence. In the initialization phase, all observed sequences are first randomly divided into segments. Segments $\boldsymbol{x}_{nj} (j = 1, 2, \cdots, J_n)$ in observed sequence $\boldsymbol{s}_n$ are then removed from the learning data, and parameter $\boldsymbol{X}_c$ of the Gaussian process and transition probability $P(c|c')$ of HSMM are updated. Segments $\boldsymbol{x}_{nj} (j = 1, 2, \cdots, J_n)$ and their classes $c_{nj} (j = 1, 2, \cdots, J_n)$ are then estimated as follows:

$$(\boldsymbol{x}_{n1}, \cdots, \boldsymbol{x}_{nJ_n}), (c_{n1}, \cdots, c_{nJ_n}) \sim P(\boldsymbol{X}, \boldsymbol{c}|\boldsymbol{s}_n), \tag{7}$$

where $\boldsymbol{X}$ is a set of segments into which $\boldsymbol{s}_n$ is divided, and $\boldsymbol{c}$ denotes classes of the segments. To carry out this sampling efficiently, the probability of all possible segments $\boldsymbol{X}$ and classes $\boldsymbol{c}$ must be computed; however, these probabilities are difficult to compute simply because the number of potential combinations is very large. Thus, we utilize forward filtering-backward sampling, which we presently explain. After sampling $\boldsymbol{x}_{nj}$ and c_{nj}, parameter $\boldsymbol{X}_c$ of the Gaussian process and transition probability $P(c|c')$ of HSMM are updated by adding them to the learning data. The segments and parameters of Gaussian processes are optimized alternately by iteratively performing the above procedure. Algorithm ?? shows the pseudocode of the blocked Gibbs sampler. $N_{c_{nj}}$ and $N_{c_{nj}, c_{n,j+1}}$ represent parameters for computing the transition probability in Eq. (??).

3.2.2 Forward filtering-backward sampling

In this study, we regard segments and their classes as latent variables that are sampled by forward filtering-backward sampling. In forward filtering, as shown in Fig. ??, the probability that k samples

Algorithm 1 Blocked Gibbs Sampler

```
// Iterate the following procedure until convergence
for $n = 1$ to $N$ do
    for $j = 1$ to $J_n$ do
        $N_{c_{nj}} - = 1$
        $N_{c_{nj}, c_{n,j+1}} - = 1$
        Delete segments $\boldsymbol{x}_{nj}$ from $\boldsymbol{X}_{c_{nj}}$
    end for

    // Sample segments and their classes
    $(\boldsymbol{x}_{n1}, \cdots, \boldsymbol{x}_{nJ_n}), (c_{n1}, \cdots, c_{nJ_n}) \sim P(\boldsymbol{X}, \boldsymbol{c} | \boldsymbol{s}_n)$

    for $j = 1$ to $J_n$ do
        $N_{c_{nj}} + +$
        $N_{c_{nj}, c_{n,j+1}} + +$
        Add segments $\boldsymbol{x}_{nj}$ into $\boldsymbol{X}_{c_{nj}}$
    end for
end for
```

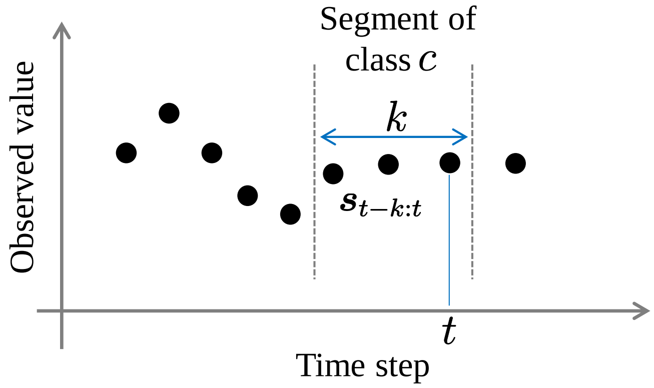


Figure 5. A segment whose probability is computed during forward filtering.

$\boldsymbol{s}_{t-k:t}$ prior to time step t in observed sequence $\boldsymbol{s}$ form a segment, and that the resulting segment belongs to class c, is computed as follows:

$$\alpha[t][k][c] = P(\boldsymbol{s}_{t-k:t} | \boldsymbol{X}_c) \times \sum_{k'=1}^{K} \sum_{c'=1}^{C} p(c|c') \alpha[t-k][k'][c'], \tag{8}$$

where C and K denote the number of classes and the maximum length of segments, respectively. $P(\boldsymbol{s}_{t-k:t} | \boldsymbol{X}_c)$ represents the probability that $\boldsymbol{s}_{t-k:t}$ is generated from a class c; this is computed as follows:

$$P(\boldsymbol{s}_{t-k:t} | \boldsymbol{X}_c) = \mathcal{GP}(\boldsymbol{s}_{t-k:t} | \boldsymbol{X}_c) P_{len}(k|\lambda). \tag{9}$$

where $P_{len}(k|\lambda)$ represents a Poisson distribution with a mean parameter λ; this corresponds to the distribution of the segment lengths. $p(c|c')$ in Eq. (**??**) represents a transition probability computed as

Algorithm 2 Forward filtering-backward sampling

```
// Forward filtering
for $t = 1$ to $T$ do
    for $k = 1$ to $K$ do
        for $c = 1$ to $C$ do
            Compute $\alpha[t][k][c]$
        end for
    end for
end for

// Backward sampling
$t = T, j = 1$
while $t > 0$ do
    $k, c \sim \alpha[t][k][c]$
    $\boldsymbol{x}_j = \boldsymbol{s}_{t-k:t}$
    $c_j = c$
    $t = t - k$
    $j = j + 1$
end while
return $(\boldsymbol{x}_{J_n}, \boldsymbol{x}_{J_n-1}, \cdots, \boldsymbol{x}_1), (c_{J_n}, c_{J_n-1}, \cdots, c_1)$
```

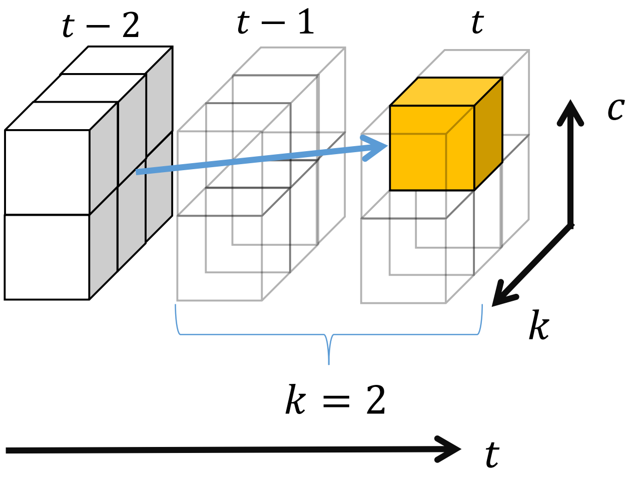


Figure 6. Recursive computation in forward filtering.

follows:

$$p(c|c') = \frac{N_{c'c} + \alpha}{N_{c'} + C\alpha}, \tag{10}$$

where $N_{c'}$ and $N_{c'c}$ denote the number of segments whose classes are c' and the number of transitions from c' to c, respectively, and k' and c' respectively denote the length and class of the segment preceding $\boldsymbol{s}_{t-k:t}$; these are marginalized out in Eq. (**??**). Moreover, $\alpha[t][k][*] = 0$ if $t - k < 0$, and $\alpha[0][0][*] = 1.0$. All elements of $\alpha[*][*][*]$ in Eq. (**??**) can be recursively computed from $\alpha[1][1][*]$ by dynamic programming. Fig. **??** depicts the computation of a three-dimensional array $\alpha[t][k][c]$. In this example, the probability

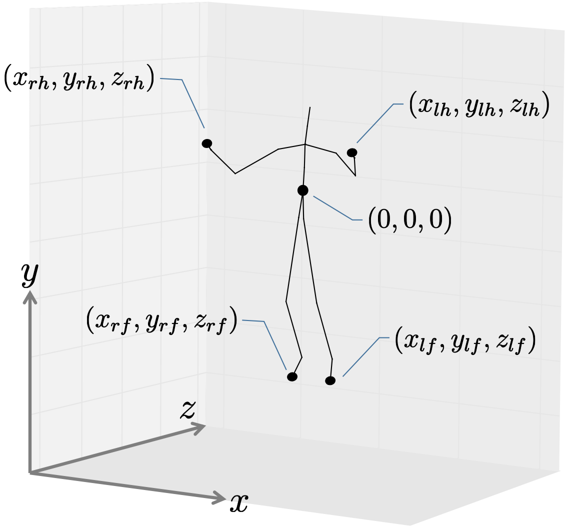

(x_{rh}, y_{rh}, z_{rh})
(x_{lh}, y_{lh}, z_{lh})
$(0, 0, 0)$
y
(x_{rf}, y_{rf}, z_{rf})
(x_{lf}, y_{lf}, z_{lf})
z
x

Figure 7. Coordinate system used in the experiments.

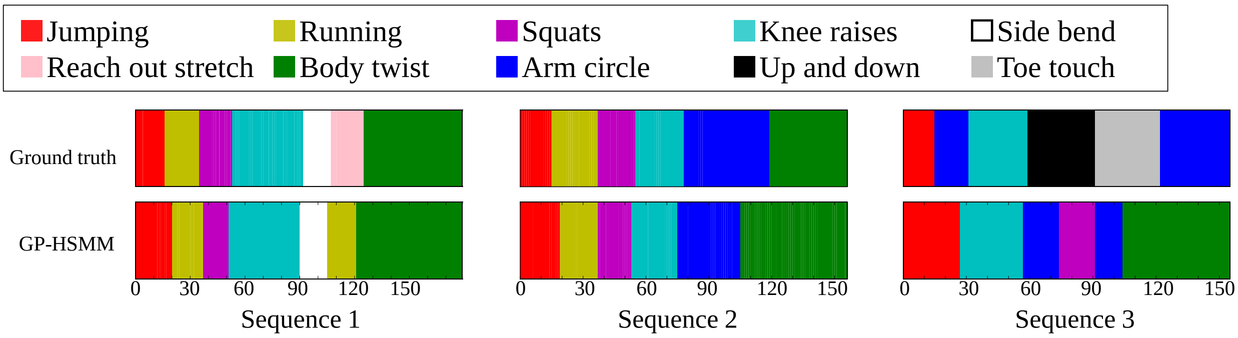


Figure 8. Segmentation results of CMU motion capture data.

that two samples before time step t become a segment is computed; the resulting segment would be assigned to class two. Hence, samples at $t-1$ and t become a segment, and all the segments whose end point is $t-2$ can potentially transit to this segment. $\alpha[t][2][2]$ can be computed by marginalizing out these possibilities.

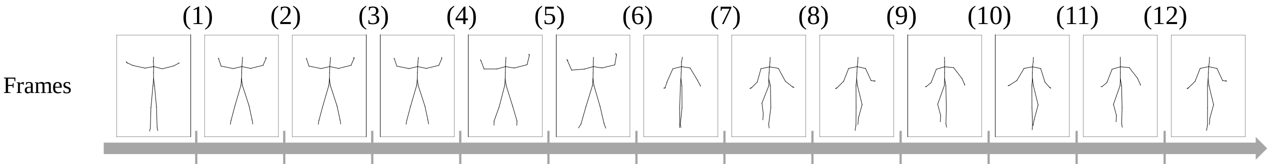
Finally, segment $\boldsymbol{x}_j$ and its class are determined by backward sampling length k and class c of the segment, based on forward probabilities in α. From $t = T$, length k_1 and class c_1 are determined according to $k_1, c_1 \sim \alpha[T][k][c]$, and $\boldsymbol{s}_{T-k_1:T}$ becomes a segment whose class is c_1. Then, length k_2 and class c_2 of the next segment are determined according to $k_2, c_2 \sim \alpha[T-k_1][k][c]$. By iterating this procedure until $t = 0$, the observed sequence can be divided into segments and their classes can be determined.

4 EXPERIMENTS

We conducted experiments to confirm the validity of the proposed method. We used two types of motion capture data: 1) data from the CMU motion capture dataset (**?**), and 2) data containing karate motions.

Table 1. Segmentation accuracy of CMU motion capture data.

Hamming distance	Precision	Recall	F-measure
0.33	0.81	0.81	0.81



Ground truth boundaries	O	X	X	X	X	O	X	X	X	X	X	X
Estimated boundaries	X	O	X	X	X	X	X	X	X	X	O	X
Evaluation of boundaries	TN	TP	TN	TN	TN	FN	TN	TN	TN	TN	FP	TN

Figure 9. Example of segmentation evaluation. Estimated boundaries are evaluated as true positive (TP), true negative (TN), false positive (FP), or false negative (FN).

4.1 Segmentation of exercise motions

We first applied our proposed method to CMU motion capture data containing several exercise routines. The CMU motion capture data was captured using a Vicon motion capture system, and positions and angles of 31 body parts are available. The dataset contains 2605 trials in six categories and 23 subcategories, and motions in each subcategory were performed by one or a few subjects. In this experiment, three sequences from subject 14 in the general exercise and stretching category were used, and include running, jumping, squats, knee raises, reach out stretches, side stretches, body twists, up and down movements, and toe touches. To reduce computational cost, we downsampled from 120 frames per second to 4 frames per second. Fig. **??** shows the coordinate system of motion capture data used in this experiment; two-dimensional frontal views of the left hand (x_{lh}, y_{lh}), right hand (x_{rh}, y_{rh}), left foot(x_{lf}, y_{lf}), and right foot (x_{rf}, y_{rf}) were used. Therefore, each frame was represented by eight dimensional vectors: $(x_{lh}, y_{lh}, x_{rh}, y_{rh}, x_{lf}, y_{lf}, x_{rf}, y_{rf})$. Because GP-HSMM requires the number of classes to be specified in advance, we set it to 8.

Fig. **??** shows the results of the segmentation. The horizontal axis represents the frame number, and the colors represent motion classes into which each segment was classified. The segments were classified into seven classes out of eight. Table 1 shows the accuracy of the segmentation. We computed the following normalized Hamming distance between the unsupervised segmentation and the ground truth:

$$ND(\boldsymbol{c}, \bar{\boldsymbol{c}}) = \frac{D(\boldsymbol{c}, \bar{\boldsymbol{c}})}{|\bar{\boldsymbol{c}}|}, \tag{11}$$

where $\boldsymbol{c}$ and $\bar{\boldsymbol{c}}$ represent sequences of estimated motion classes and true motion classes, $D(\boldsymbol{c}, \bar{\boldsymbol{c}})$ is the Hamming distance between two sequences, and $|\bar{\boldsymbol{c}}|$ represents the length of the sequence. Therefore, the normalized Hamming distance ranges from 0 to 1; lower Hamming distances indicate more accurate segmentation. In this experiment, the Hamming distance was 0.33, which is comparable with the BP-HMM reported in (**?**). However, they also reported that some segments were split into two or more categories, and that those shorter segments were bridged. In contrast, we performed no such modifications, and Fig. **??** shows that there are no shorter segments. We also computed the precision, recall, and F-measure of the segmentation. To compute them, estimated boundaries of segments are evaluated as true

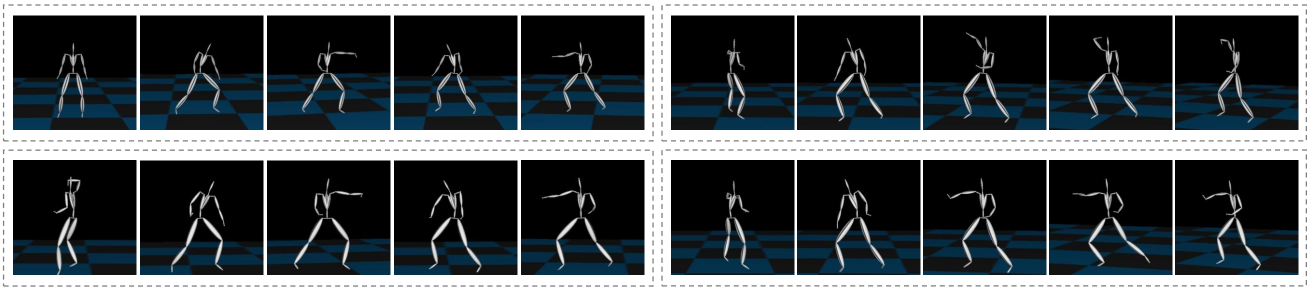


Figure 10. Motion capture data of karate motions.

Table 2. Segmentation accuracy of karate motions.

	Hamming distance	Precision	Recall	F-measure
GP-HSMM	0.21	0.92	0.92	0.92
HDP-HMM	0.47	0.12	0.54	0.19
HDP-HMM + NPYLM	0.61	0.00	0.00	0.00
BP-HMM	0.49	0.13	0.23	0.16
AutoPlait	0.76	0.00	0.00	0.00

positive (TP), true negative (TN), false positive (FP), or false negative (FN). Fig. **??** shows an example of segmentation evaluation. We considered the estimated boundary to be TP if it was within true boundary ± four frames, as shown in Fig. **??**(2). If the ground truth boundary has no corresponding estimated boundary as shown in Fig. **??**(6), it was considered as FN. Conversely, if the estimated boundary has no corresponding ground truth boundary as shown in Fig. **??**(11), it was considered as FP. From these evaluations, the precision, recall, and F-measure of the segmentation are computed as follows:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}}, \tag{12}$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}}, \tag{13}$$

$$F = \frac{2PR}{P + R}, \tag{14}$$

where N_{TP}, N_{FP}, and N_{TN} represent the number of points assessed as TP, FP, and FN. The F-measure of the segmentation was 0.81, and this fact indicates that GP-HSMM can estimate boundaries reasonably. This is because GP-HSMM estimates the length of segments as well as the classes of segments.

Moreover, Fig. **??** shows that most false segmentations are in sequence 3. This is because "up and down" and "toe touch" motions are included only in sequence 3, and GP-HSMM was not able to extract patterns that occur infrequently. However, this problem is not limited to GP-HSMM, and it is generally difficult for any learning method to extract infrequent patterns. The Hamming distance, which was computed only from sequence 1 and sequence 2, was 0.15. This result shows that GP-HSMM can accurately estimate segments that appear multiple times in a sequence.

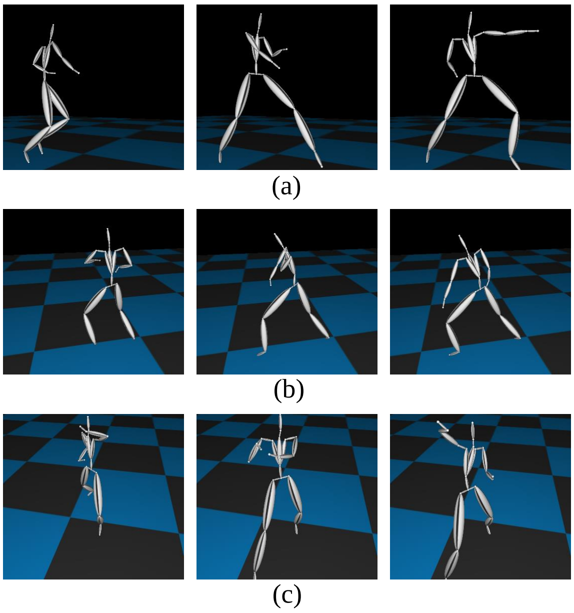


Figure 11. Basic motions in Kata: (a) Left punch. (b) Left lower guard. (c) Right upper guard.

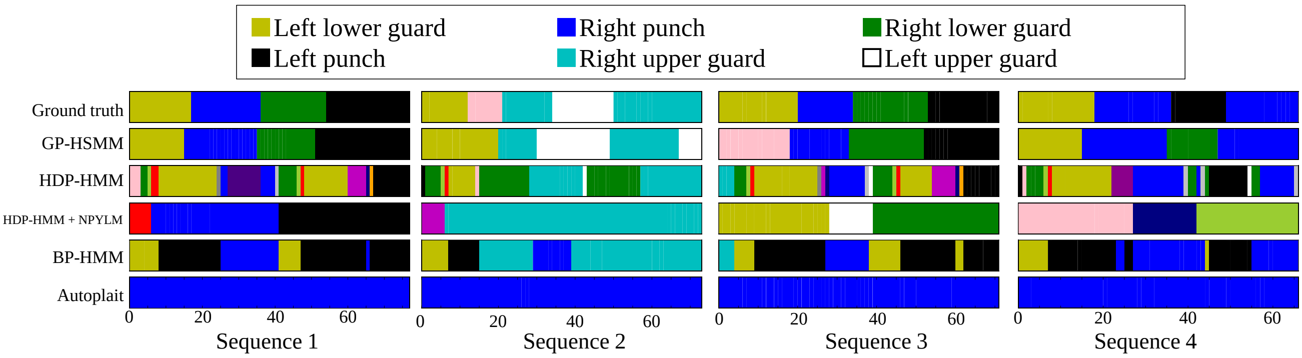


Figure 12. Results of segmentation and classification for each method.

4.2 Segmentation of karate motion

We then applied our proposed method to more complex motion capture data, which consisted of the basic motions of karate (called kata in Japanese) [1] （Fig. **??**） from the motion capture library Mocapdata.com[2]. There are fixed motion patterns (punches or guards) in kata, and it is easy to form a ground truth for the segmentation. However, there might be shorter motion patterns, and GP-HSMM might be able to find those motion patterns if the number of classes is set to a larger number. Moreover, it is possible for GP-HSMM to discover patterns that cannot be labeled by humans, and GP-HSMM has the potential to analyze unlabeled time series data. However, in this experiment, we must evaluate the proposed method quantitatively, and fixed motion patterns (punches or guards) labeled by a human expert are used as ground truth. The type of kata we used was called heian 1, which is the most basic form of kata consisting of punches, lower guard, and upper guard (Tsuki, Gedanbarai, and Joudanuke in Japanese). Fig. **??** shows

[1] http://mocapdata.com/product.cgi?product_id=10019

[2] http://www.mocapdata.com/

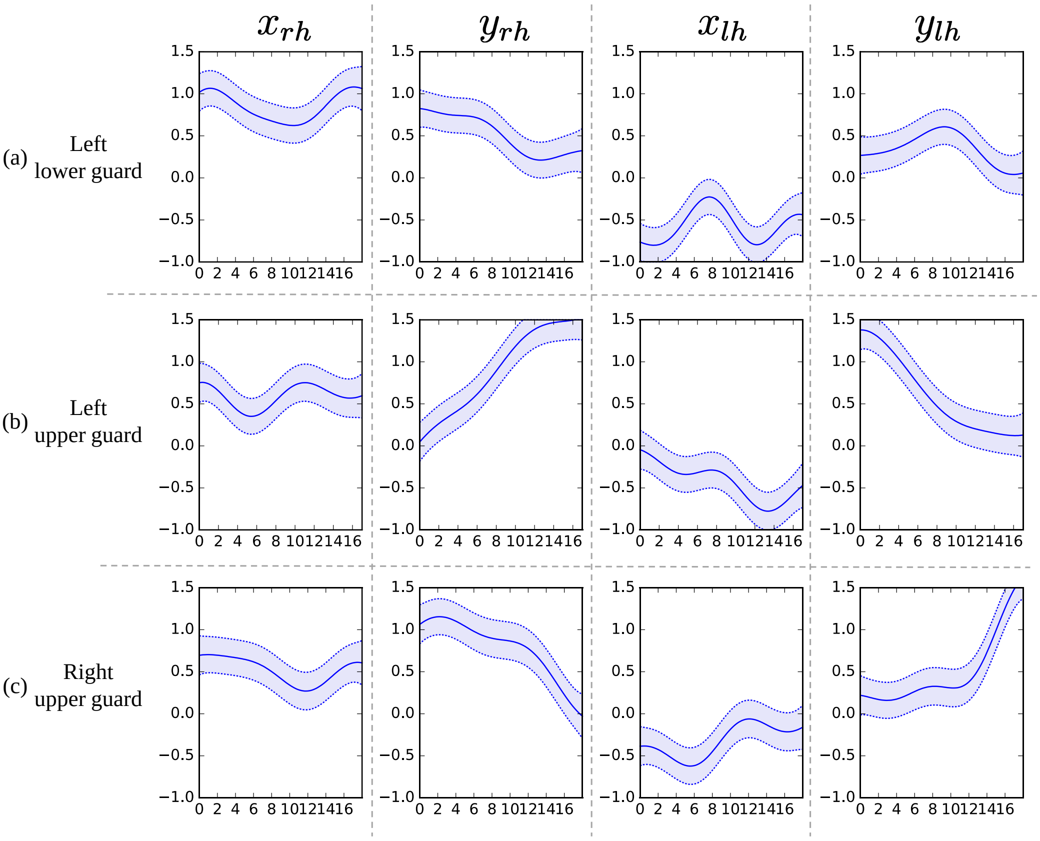


Figure 13. Learned Gaussian processes for left lower guard, left upper guard, and right upper guard.

the basic movements used in heian 1. We divided this motion sequence into four parts, for use as four motion sequences to apply the blocked Gibbs sampler. Each motion sequence consisted of the following actions:

1. Left lower guard, right punch, right lower guard, and left punch.
2. Left lower guard, right upper guard, left upper guard, and right upper guard.
3. Left lower guard, right punch, right lower guard, and left punch.
4. Left lower guard, right punch, left punch, and right punch

By way of its preprocessing, as shown in Fig. **??**, the motion capture data was converted into motions with the body facing forward with a center of (0,0,0). To reduce computational cost, we downsampled the motion capture data from 30 frames per second to 15 frames per second, and used two-dimensional left-hand positions (x_{lh}, y_{lh}) and right-hand positions (x_{rh}, y_{rh}) in the frontal view, as shown in Fig. **??**. To compare our method with others, we used segmentation based on HDP-HMM (**?**) and segmentation based on NPYLM and HDP-HMM (**?**), where NPYLM (**?**) divides sequences discretized by HDP-HMM. In addition, we compared our method with BP-HMM (**?**) and AutoPlait (**?**).

Fig. **??** shows the segmentation results. The horizontal axis represents the frame number, and the colors represent motion classes into which each segment was classified. The figure shows that HDP-HMM estimated shorter segments than the ground truth. This occurred because the emission distribution of HDP-HMM is a Gaussian distribution, which cannot represent continuous trajectories. Moreover, the result produced by segmentation, in which NPYLM divided sequences discretized by HDP-HMM, yielded

Table 3. Computational time of each method.

	time (sec)
GP-HSMM	248
HDP-HMM	1.99
HDP-HMM + NPYLM	18.2
BP-HMM	3.37
AutoPlait	0.31

longer segments. Moreover, NPYLM cannot extract fixed patterns of sequences. This is because the sequences discretized by HDP-HMM included noise and, therefore, NPYLM was unable to find a pattern in them. It was also difficult for BP-HMM to estimate correct segments, and some shorter segments were present. Further, AutoPlait could not find any segments in the karate motion sequences. We believe this occurred because HMMs are too simple to model complex motions. On the contrary, we use Gaussian processes that make it possible to model complex sequences. Table **??** shows the segmentation accuracy of each method. We considered the estimated boundary to be correct if it was within true boundary ± five frames. The F-measure of the proposed method was 0.92, which indicates that GP-HSMM can estimate boundaries accurately. The results show that GP-HSMM outperforms the other methods. Fig. **??** shows the learned Gaussian process. y_{rh} in Fig. **??**(a), which represents the height of the left hand, is decreased, which indicates the motion where the left hand is dropped for the lower guard. In contrast, y_{rh} in Fig. **??**(b) is increased, which indicates the motion where the left hand is raised for the upper guard. Conversely, y_{lh} in Fig. **??**(c) is increased for the right upper guard. From this result, we can see that characteristics of motions can be learned by Gaussian processes.

Moreover, the motions were classified into seven classes, although we set the number of classes to eight. This result indicates that the number of classes can be estimated to a certain extent, if a number closer to the correct number is given. However, a smaller number leads to under-segmentation and misclassification, and a much larger number leads to over-segmentation. This is a limitation of the current GP-HSMM, and we believe it can be solved by introducing a non-parametric Bayesian model.

Computational cost is another limitation of GP-HSMM. Table **??** shows the computational time required to segment karate motion. HMM-based methods such as HDP-HMM, BP-HMM, and AutoPlait are relatively faster. In particular, AutoPlait is the fastest because it uses a single scan algorithm proposed in (**?**) to find boundaries, and it has been demonstrated that AutoPlait can detect meaningful patterns from large datasets. In contrast, our proposed GP-HSMM is much slower than other methods, and cannot process such large datasets. This is another limitation of the proposed method.

5 CONCLUSION

In this paper, we proposed a method for motion segmentation based on a hidden semi-Markov model (HSMM) with a Gaussian process (GP) emission distribution. By employing HSMM, segment classes and their lengths can be estimated. Moreover, a forward filtering-backward sampling algorithm is used to estimate the parameters of GP-HSMM; this makes it possible to efficiently search for all possible segment lengths and classes. The experimental results showed that the proposed method can accurately segment motion capture data. Although motions that occurred in the sequences a single time were difficult to segment correctly, motions that occurred a few times could be segmented with higher accuracy.

However, some issues remain in the current GP-HSMM. The most significant problem is that GP-HSMM requires the number of classes to be specified in advance. We believe this value can be estimated by utilizing a non-parametric Bayesian model. We are planning to introduce a stick-breaking process as a prior distribution of the transition matrix, and beam sampling for parameter estimation; these techniques are utilized in (**?**). Another problem is computational cost. The computational cost to learn a Gaussian process is $O(n^3)$, where n denotes the number of data points classified in the GP. To overcome this problem, efficient computation methods have been proposed (**??**), and we will consider introducing these methods into GP-HSMM.

ACKNOWLEDGEMENT

This work was supported by JST CREST Grant Number JPMJCR15E3 and JSPS KAKENHI Grant Number JP17K12758.

REFERENCES

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems* 57, 469–483

Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. (2001). The infinite hidden markov model. In *Advances in neural information processing systems*. 577–584

CMU (2009). CMU graphics lab motion capture database. `http://mocap.cs.cmu.edu/`

Fod, A., Matarić, M. J., and Jenkins, O. C. (2002). Automated derivation of primitives for movement classification. *Autonomous Robots* 12, 39–54

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2007). The sticky hdp-hmm: Bayesian nonparametric hidden markov models with persistent states. *MIT Laboratory for Information and Decision Systems Technical Report 2777*

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2011). Joint modeling of multiple related time series via the beta process. *arXiv preprint arXiv:1111.4226*

Goldwater, S. (2006). *Nonparametric Bayesian Models of Lexical Acquisition*. Ph.D. thesis, Brown University

Gräve, K. and Behnke, S. (2012). Incremental action recognition and generalizing motion generation based on goal-directed features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 751–757

Lin, J. F.-S., Karg, M., and Kulić, D. (2016). Movement primitive segmentation for human motion modeling: A framework for analysis. *IEEE Transactions on Human-Machine Systems* 46, 325–339

Lin, J. F.-S. and Kulić, D. (2012). Segmenting human motion for automated rehabilitation exercise analysis. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2881–2884

Lioutikov, R., Neumann, G., Maeda, G., and Peters, J. (2015). Probabilistic segmentation applied to an assembly task. In *IEEE-RAS International Conference on Humanoid Robots*. 533–540

Manschitz, S., Kober, J., Gienger, M., and Peters, J. (2015). Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations. *Robotics and Autonomous Systems* 74, 97–107

Matsubara, Y., Sakurai, Y., and Faloutsos, C. (2014). Autoplait: Automatic mining of co-evolving time sequences. In *ACM SIGMOD International Conference on Management of Data*. 193–204

Mochihashi, D., Yamada, T., and Ueda, N. (2009). Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling. In *Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing*. vol. 1, 100–108

Nguyen-Tuong, D., Peters, J. R., and Seeger, M. (2009). Local gaussian process regression for real time online model learning. In *Advances in Neural Information Processing Systems*. 1193–1200

Okadome, Y., Urai, K., Nakamura, Y., Yomo, T., and Ishiguro, H. (2014). Adaptive lsh based on the particle swarm method with the attractor selection model for fast approximation of gaussian process regression. *Artificial Life and Robotics* 19, 220–226

Shiratori, T., Nakazawa, A., and Ikeuchi, K. (2004). Detecting dance motion structure through music analysis. In *IEEE International Conference on Automatic Face and Gesture Recognition*. 857–862

Takano, W. and Nakamura, Y. (2016). Real-time unsupervised segmentation of human whole-body motion and its application to humanoid robot acquisition of motion symbols. *Robotics and Autonomous Systems* 75, 260–272

Taniguchi, T. and Nagasaka, S. (2011). Double articulation analyzer for unsegmented human motion using pitman-yor language model and infinite hidden markov model. In *IEEE/SICE International Symposium on System Integration*. 250–255

Uchiumi, K., Hiroshi, T., and Mochihashi, D. (2015). Inducing Word and Part-of-Speech with Pitman-Yor Hidden Semi-Markov Models. In *Joint Conference of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. 1774–1782

Wächter, M. and Asfour, T. (2015). Hierarchical segmentation of manipulation actions based on object relations and motion characteristics. In *International Conference on Advanced Robotics*. 549–556