Graph-Based Action Models for Human Motion Classification

Felix Endres Jürgen Hess Wolfram Burgard
University of Freiburg, Dept. of Computer Science, Freiburg, Germany

Abstract

Recognizing human actions is an important ability for service and domestic robots. This paper presents a novel approach for learning and recognizing motion models from human motion capturing data. The key idea is to represent observed motion trajectories as a graph, where the nodes correspond to poses and the edges indicate pose similarities. We optimize this graph using least squares minimization and non-maximum suppression to obtain a generalized model for the respective action. The resulting motion models can then be used to recognize actions in unlabeled motion capturing data. Experiments based on real-world data show that the learned motion models can reliably classify a large set of different motions. Furthermore, we show that the learned models robustly generalize over different people.

1 Introduction

Observing and interpreting human motion is an important prerequisite for many applications involving human-robot interaction. Gesture recognition, for example, allows a robot to classify the actions of an operator and act accordingly. Recognizing human actions can furthermore provide valuable information about the environment and objects people use. E.g., if someone is observed drinking, the probability that the object held by the person is a cup, a glass, or a bottle is high.

What makes the classification of human motion hard is that the continuous and often high-dimensional motion data needs to be classified into a small set of discrete actions. The difficulty of building such a model is further increased by the large variations in human motion, both for individual persons, and over multiple people.

Our approach derives generalized patterns from motion data which are accurate representatives of the demonstrated actions. It extracts generalized, compact action models from labeled training data. We capture motion data using the full-body motion capturing suit shown in **Figure 1**, to track the position of 23 body segments with 120 Hz. We perform a principal component analysis (PCA) to reduce the dimensionality of the data. By computing the distance between the time derivatives of the poses, we construct an undirected graph. Each node of the graph represents a human pose and the edges denote associations between poses. First, we use least-squares minimization on the graph. Second, we extract a representative action model. This step largely reduces the number of poses required to describe a model.

The main contribution of this paper is the graph representation for motion trajectories and the extraction of representative trajectories for that motion which can be used in a variety of applications. The presented approach neither requires the training instances to be segmented nor assumes the association of corresponding poses to be given. It allows for reliable classification of different motions, even if they have similar trajectories and deals with low and high frequency motions. We evaluate our approach in a supervised classification setting with five different action types. To demonstrate the capabilities we chose actions with similar trajectories (walking, jogging, walking up and down staircases, and riding a bicycle) and present experiments about the generalization over multiple people.

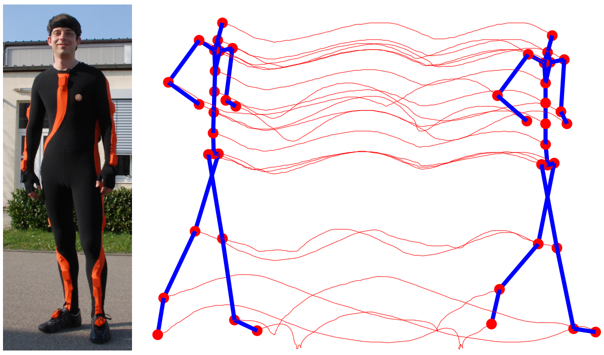


Figure 1: Left: Sensor equipment used for data acquisition. Right: Typical motion trajectories of a jogging step.

2 Related Work

There is a large body of literature that is concerned with activity recognition, selection of relevant features for motion recognition, and segmentation of human motion or time series data in general. A comprehensive overview is provided by Preece *et. al* [8] and Aggarwal and Park [1]. In this section we only review approaches that use temporal models over several poses for motion segmentation and classification as those are more closely related to our approach.

A common technique for modeling temporal data are Hidden Markov Models (HMMs) [9] which also have been applied to classification and segmentation of motion capturing data. Cielniak *et. al* [2] use the EM algorithm to cluster low dimensional motion trajectories and find the most likely generalization for each cluster. An HMM is created from the generalization and used to predict future behavior. In contrast to our work, their method requires the segmentation of the trajectory instances to be given and assumes that movements are homogeneous in velocity.

Kohlmorgen *et. al* [3] propose a system for unsupervised motion segmentation based on HMMs. They assume that the data belonging to the same segment is underlying the same probability distribution and describe motion sequences as a series of probability density functions which are used as states in an HMM. Computing the most likely path in the HMM then results in the segmentation of the data. Kulic *et. al* [5] construct an HMM model for the observed motions and define a distance function for HMMs. Based on this distance function they use hierarchical clustering for grouping the HMMs. The disadvantage of HMMs is that one has to specify the number of states in advance which is a hard task considering the different time scales and durations of human actions. Additionally, the computational cost for learning the transition weights of the HMMs are typically high.

A common technique for dealing with the wide range of time scales is Dynamic Time Warping (DTW) which has also been used in the area of motion segmentation and classification. Often, DTW serves as a preprocessing step for aligning different time series to each other. In the area of motion classification Müller *et. al* [7] apply DTW as preprocessing of time series data consisting of a set of binary features and use the result for specifying motion templates. In this paper we apply an approach related to DTW. While DTW calculates the similarity of one frame to the others to compute a distance measure, we use a neighborhood of several frames to compute the similarity. For our data we found our approach more reliable. The approach most similar to ours for finding similarities between different time series is the one by Zhou *et. al* [14]. They apply the Dynamic Time Alignment Kernel (DTAK) for time-invariant alignment of motion sequences and then use k-Means clustering to segment motion capture data into actions. However, they do not use the gained information to construct representative models of human actions.

A recent method for modeling human motion data has been developed by Wang *et. al* [12]. They propose a two stage process learning a low dimensional data representation and a motion model in the latent space using a Gaussian process latent variable model. This approach has been extended to motion classification by Raskin *et. al* [10]. Gaussian processes however have high computational costs and thus only few data points can be used.

Our approach uses the principal component analysis (PCA) to reduce the dimensionality of the data, which allows for efficient mapping of new data. The full number of training samples is only necessary during the learning phase. The resulting models are compact and can serve as a template for action classification similar to the templates constructed in Müller *et. al* [7]. However, in difference to Müller *et. al*, we use continuous features of free-form motion trajectories that are automatically generated from the training data. This removes the requirement to manually select features. Representing motions in a graph structure has been explored by Kovar *et. al* [4] and Lee *et. al* [6]. Both groups aim to synthesize realistic motions from a database of captured motion segments. In both works, the graph is used to search for a motion path that meets the requirements of the motion to be generated. In contrast, we use the graph representation to find a generalization over recurring motion patterns. Yamane *et. al* [13] recursively cluster the poses to construct a binary tree. Several graphs are then computed by connecting the clusters of each level with edges weighted by the transition frequencies between the contained poses.

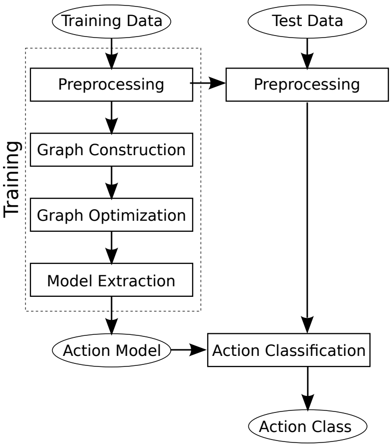


Figure 2: Model construction and action classification.

3 Methodology

This section describes our approach to the construction of action models and its application to a supervised classification task. We start with a description of the input data and the applied dimensionality reduction technique in Section 3.1. We then define a similarity measure for short motion segments. We use it to construct a graph from the preprocessed motion trajectories by interpreting the poses as nodes and introducing edges according to their time sequence and similarity (Section 3.2). These two types of edges represent conflicting spatial information. By defining an error measure on the graph edges and applying least squares optimization we find a trajectory that minimizes the inconsistency (Section 3.4). The resulting trajectory serves as the basis for computing a compact representative model for the diverse motions patterns. The individual steps of our approach are depicted in **Figure 2**.

3.1 Data Preprocessing

Motion capturing of an entire body pose typically results in high dimensional data. In our experiments the input data consists of 23 joint positions given by their three Cartesian coordinates. Thus the data is 69 dimensional. We first transfer the data to a local reference frame and then reduce its dimensionality using PCA. The PCA has several advantages for our application. First, it is a linear projection and preserves the continuity of the motion capturing data in the low dimensional subspace. Second, the projection to the subspace and its inverse are readily available, and applicable to new data. Third, the PCA allows us to choose the subspace, such that the predominant motion information is retained. Based on the assumption that the motion pattern to be learned dominates the variance in the training data set, the dimensions corresponding to high eigenvalues capture relevant parts of the target pattern. Likewise, dimensions which are not characteristic are likely to be represented by dimensions with smaller eigenvalues. We select a minimal number of eigenvectors whose cumulated eigenvalues exceeded 95 % of the sum of all eigenvalues.

3.2 Similarity Measure

Having projected the data to a low dimensional subspace the goal is to compute representative models for the captured motions in the form of repetitive patterns. In the case of repetitive motions like walking there typically are many instances of such patterns. However, we found such patterns in most of the activities that last over longer periods of time. A key requirement to learn such a generalization based on repetitive patterns is to find associations between poses that constitute corresponding parts of the motion.

A single posture is not descriptive for a motion. To find similar motions Kovar *et. al* [4] therefore use a similarity measure that incorporates the temporal neighborhood of the poses in the distance calculation. They first compute a rigid transformation in the horizontal plane to align the motion segments and then compute the squared error between the poses based on point displacements. Lee *et. al* [6] determine a weighted sum over the squared joint angle differences with manually selected weights.

We also determine associations based on the pairwise similarity of poses within a neighborhood of k poses, but compare the direction of the motion directly. For aligning the postures, we represent all coordinates relative to the pose of the hip joint and let the dimensionality reduction select the influence of the individual joints. More precisely, we compute the similarity of a pair of poses x_i and x_j by comparing the respective time sequence of $2 \cdot k + 1$ poses centered at x_i and x_j respectively. We define the similarity $s(x_i, x_j)$ as

$$s(x_i, x_j) = \frac{1}{2k+1} \sum_{q=-k}^{k} \frac{< x'_{i+q}, x'_{j+q} >}{|x'_{i+q}| \cdot |x'_{j+q}|}. \qquad (1)$$

The similarity is therefore defined as the mean over the dot product (denoted by $<\cdot,\cdot>$) of the normalized directional derivatives of the segments located at poses (x_{i+q}, x_{j+q}), where q denotes all integer values in range $(-k \dots k)$. x'_{i+q} and x'_{j+q} denote the directional derivative at points x_{i+q} and x_{j+q}. The derivative is computed by fitting a cubic spline to the poses of the segment and obtaining its derivative at the desired position. The spline also provides the possibility to abstract from time dependencies, i.e., the velocity of the motion, as it allows us to resample the poses at every desired time scale or equidistant with respect to the trajectory length (in contrast to the “equi-timed” original data).

3.3 Graph Construction

Calculating the similarity between each pair of poses results in a similarity matrix S where the entry at row i and column j are specified as the similarity $s(x_i, x_j)$. To extract the associations from the similarity matrix, we only consider local maxima above a threshold γ, which results in a sparse association matrix. Interpreting the poses as nodes and the associations as edges, this matrix can be seen as a graph. Because we need to compute the pairwise similarity, the computational complexity of the graph construction is quadratic in the number of pose samples. However, as we build one graph per action type, the quadratic runtime is a restriction on the number of samples per action, whereas the runtime only increases linearly with number of actions. The computation of the similarity matrix can easily be parallelized, as the individual similarity computations are independent of each other.

Having obtained this graph one approach could be to average over the positions of the connected nodes to find a trajectory that is a representative for all the others. This method, however, ignores the overall shape of the motion trajectory and typically leads to models that are greatly distorted. To incorporate the shape of the trajectory, we use the translation between successive poses to extend the graph with edges.

3.4 Graph-Based Trajectory Optimization

Given the graph, we seek the trajectory that generalizes over the corresponding poses, while taking the overall shape of the motion into account. Due to the high variability of human motion data there are many spurious and missing associations. Therefore, an approach that requires consistent correspondences to be given, e.g., [2], is not applicable. In order to maintain the overall shape of the trajectory, we associate spatial information z to each edge, which represents the desired translation between the respective nodes.

Graph edges that connect consecutive nodes i and $i+1$, are associated with the respective translational difference $z_{i\,i+1} = (x_{i+1} - x_i)$, i.e., the original motion between time steps i and $i+1$. To express that corresponding nodes should be unified, we associate the graph edges between corresponding nodes with a column vector $z_{ij} = \vec{0}$. The resulting graph consists of n nodes and m edges, where m is the number of correspondences plus the number of time steps.

As the information of the two types of edges conflicts, we apply least squares optimization on the graph, to find a trajectory that minimizes the squared error with respect to the spatial information of the edges. This approach is similar to the methods used for solving graph-SLAM [11]. Using only translational information allows for a linear solution that can be computed efficiently. We define the error of the graph with nodes $\mathbf{x}$ and edges $\mathbf{z}$ to be

$$F(\mathbf{x}, \mathbf{z}) = \sum_{ij \in E} \omega_{ij}(z_{ij} - (x_i - x_j))^T (z_{ij} - (x_i - x_j)) \quad (2)$$

where, the $n \times d$ Matrix $\mathbf{x} = [x_1 \cdots x_n]^T$, contains the d-dimensional position vectors of the graph nodes, and the $d \times m$ Matrix $\mathbf{z} = [\cdots z_{ij} \cdots]$, contains the translation vectors of the graph edges and ω_{ij} is a scalar used to weight the edges. E represents the set of all edges.

To find the trajectory $\hat{\mathbf{x}}$ with minimum error, we set the derivative of the error function $F(\mathbf{x}, \mathbf{z})$ with respect to $\mathbf{x}$ to zero. As $F(\mathbf{x}, \mathbf{z})$ is quadratic, the solution of the linear system yields the global minimum. The final result is the trajectory with the minimal squared error with respect to the error function defined, i.e., a trajectory minimizing the distance of corresponding points with minimal dislocation from their original positions. The effect of this procedure is depicted in **Figure 3**. **Figure 4(a)** and **Figure 4(b)** visualize the first three principal components of real training data during the optimization steps. It is clearly visible that the trajectory exhibits a greatly enhanced homogeneity.

3.5 Model Extraction

The least-squares optimization leads to motion trajectories which are more homogeneous but does not reduce the number of data points. To efficiently use the new motion trajectory for classification of new motions, we want to extract models with reduced redundancy. To achieve this, we merge nodes based on their correspondences. In our experiments, we found that the associations between the nodes improve substantially if they are recomputed given the homogenized trajectory. To merge nodes, we then compute the mean pose of the associated nodes for each node with a sufficient amount of correspondences ϕ. In our experiments we set $\phi = 3$ but the exact value is not critical.

As not the entire graph might be connected, the model extraction returns one model for each connected component of the graph. Thus, the model generation algorithm also has the advantage that the number of necessary model trajectories to represent the input data is found automatically and does not need to be specified in advance. Finally, we discard models that are too short and thus do not describe useful motions. Due to the compactness of the resulting model trajectories, it possible to efficiently compare them to new motions. **Figure 4(c)** shows the result of model extraction. Note that in contrast to the data shown in **Figure 4(b)** the number of poses has been largely reduced.

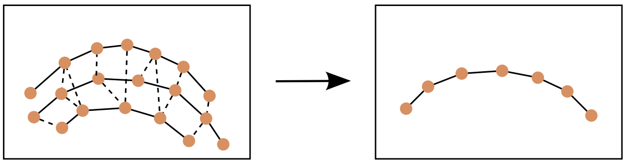


Figure 3: The left diagram represents a graph as constructed from the motion trajectories. The solid edges connect successive nodes while the dashed edges were introduced due to their similarity. The right picture shows the desired result of the optimization.

3.6 Supervised Motion Classification

In this section we describe the application of our approach to supervised motion classification which we evaluate in Section 4. In a supervised setting the action type is known for the training data. We use the action labels to construct individual graphs for each action. More precisely, we construct a model applying the following procedure to each action which (see also **Figure 2**):

1. Preprocess the data (Section 3.1).
2. Compute the graph (Section 3.2).
3. Optimize the trajectory (Section 3.4).
4. Construct a model for each action (Section 3.5).

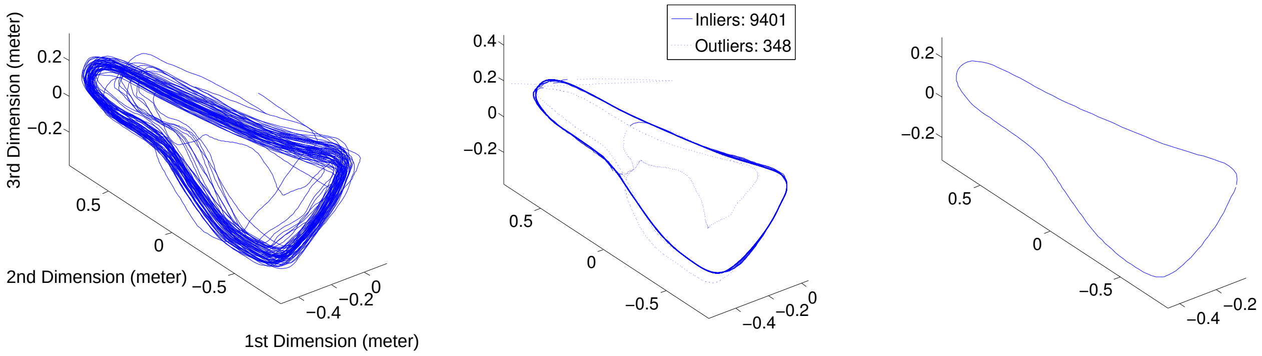
Once the models have been computed, we can use them to classify new motion data. We preprocess the test data the same way as the training data, i.e., we transfer the data from the global reference frame to the local reference frame used for the training data and apply the projection previously computed by the PCA. Thus the test data now lives in the same subspace as the training data and we can compute the similarity of the test data to each of the models computed for each action. We classify using the maximum-likelihood principle, i.e., we choose the activity for which the highest similarity is obtained.

4 Experiments

This section presents experiments, demonstrating the application of our approach for supervised classification of motion data.

4.1 Data Acquisition and Preprocessing

As described in Section 3.1 we work on the Cartesian coordinates of 23 main joints of the human skeleton. We captured this data using a suit equipped with 17 inertial measurement units (IMUs) developed by the company Xsens Technologies, shown in **Figure 1**. Using these sensors, the



(a) (b) (c)

Figure 4: (a) First principal components of the training data for walking. (b) Trajectory after application of least squares optimization. “Outliers” denotes the poses for which no correspondences could be found. (c) The resulting model, used for motion classification (contains only around 300 poses).

positions for the skeletal joints are computed with a sampling rate of up to 120 Hz. However, our approach is not specific to IMU measurements, as we rely on positional data only, which could be captured with any motion capturing system. For the experiments we considered five activities, namely walking, running, bicycle riding, and ascending and descending staircases performed by two individuals. We chose repetitive and very similar actions, e.g., jogging and walking, to demonstrate the capability of generating concise models that generalize over action cycles but are still able to discern actions very similar to each other. We use about one minute of motion capture data for each action and each of the two individuals. In total, we process about 67,000 full body poses.

The captured data was labeled and represented in the local coordinate frame of the hip. Upon applying the PCA we selected the minimal number of eigenvectors whose cumulated eigenvalues exceeded 95 % of the sum of all eigenvalues. The data was then projected to the subspace spanned by the selected eigenvectors. Using this criterion we reduced the data to eight dimensions.

4.2 Graph Construction and Trajectory Optimization

We construct a graph from the preprocessed motion data as described in Section 3.2. This requires to determine two parameters, the optimal size of the trajectory segments to be compared and the threshold we use to prune edges of the graph. The influence of the neighborhood size k is limited as long as the segments are long enough to be specific for the action type. Too long segments would result in a long transition period when the action changes. More crucial is the choice of the minimum similarity threshold, as it further sparsifies the graph (after filtering for local maxima) and, most importantly, suppresses spurious associations, which could lead to an unwanted distortion of the trajectory. This parameter also influences the generalization capabilities of the model. If a too low value is chosen, significant variations might be generalized over, making the model useless. If a too high value is chosen, slight variations will not be generalized over which results in unnecessary many, overly specific models for the same action. To determine the optimal parameter values, we applied a gradient ascent procedure, maximizing the classification performance.

After graph construction we determine the least squares solution as described in Section 3.4. The result for a dataset with two individuals descending stairs can be seen in Figure 4. We recompute the correspondence association and extract compact models as described in Section 3.5. Models shorter than the segment length are discarded, as the similarity measure can not be applied. Model construction resulted in a single trajectory, except for descending stairs, where the motion patterns of the two individuals were too distinct to allow for generalization, without loss of recognizability. The used training data sets comprise on average 6,000 pose samples. The extracted models, in contrast, typically only contain about 300 poses per action. Table 2 show the resulting pose number for the experiments presented in the next section. Note that the amount of memory required does not increase linearly with the number of training samples, but mainly depends on the number of models and the length of the motion instance to classify.

4.3 Classification Results

We evaluate our approach with two experiments. In the first experiment we split the data into a training and a test set, each of which contains the motion capture data of two people. Thus, we learn the models on both individuals and also test on both but keep training and test data separate.

In the second experiment, we split the data such that we learn the model trajectories from one individual and classify the motions of the other. This experiment aims at showing that our approach generalizes motion trajectories

Experiment	Training Set	Test Set	Correct Classif.
1	Person 1 & 2 (Fold 1)	Person 1 & 2 (Fold 2)	95.0%
	Person 1 & 2 (Fold 2)	Person 1 & 2 (Fold 1)	87,6%
2	Person 1	Person 2	95.5%
	Person 2	Person 1	88.8%

Table 1: Classification results using the learned models with disjoint training and test sets.

Action Type	Precision	Recall	Model Size
Walking	0.84	0.92	308
Jogging	0.92	0.78	193
Ascending Stairs	0.95	0.97	390
Descending Stairs	0.98	0.86	385
Bicycle Riding	0.90	0.99	243

Table 2: Classification results with respect to actions type. Values are averaged over the experiments listed in Table 1

to different people. The results are shown in Table 1 and clearly demonstrate that our approach is capable of learning motion trajectories from training data that generalize over multiple people and can be used to reliably classify human actions. Table 2 presents the precision and recall values for the individual actions, averaged over the experiments in Table 1. These statistics show that the performance is stable over all actions (with a small bias to classify jogging as walking).

5 Conclusions and Future Works

In this paper we presented a novel approach for modeling human motion trajectories using a graph representation in which the individual poses are nodes and the similarity between poses and their temporal dependencies are the edges. We use linear least squares optimization to extract compact representative trajectories for the action. These trajectories serve as free-form templates for motion classification.

We implemented and evaluated our approach on real, high-dimensional motion data. Experimental results show that it finds models that can be used for reliably classifying human actions. We also showed that models learned from one person robustly generalize to another one.

References

[1] JK Aggarwal and S. Park, *Human motion: Modeling and recognition of actions and interactions*, Proc. of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004.

[2] G. Cielniak, M. Bennewitz, and W. Burgard, *Where is ...? Learning and utilizing motion patterns of persons with mobile robots*, Proc. of the International Joint Conference on Artificial Intelligence (IJCAI), 2003.

[3] J. Kohlmorgen and S. Lemm, *A dynamic hmm for on-line segmentation of sequential data*, Advances in Neural Information Processing Systems **1** (2002), 793–800.

[4] L. Kovar, M. Gleicher, and F. H. Pighin, *Motion graphs*, SIGGRAPH, 2002, pp. 473–482.

[5] D. Kulic, W. Takano, and Y. Nakamura, *Incremental on-line hierarchical clustering of whole body motion patterns*, IEEE International Symposium on Robot and Human Interactive Communication, 2007, pp. 1016–1021.

[6] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard, *Interactive control of avatars animated with human motion data*, SIGGRAPH, 2002, pp. 491–500.

[7] M. Müller and T. Röder, *Motion templates for automatic classification and retrieval of motion capture data*, Proc. of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2006.

[8] SJ Preece, JY Goulermas, LP Kenney, D. Howard, K. Meijer, and R. Crompton, *Activity identification using body-mounted sensors-a review of classification techniques.*, Physiological Measurement **30** (2009), no. 4, R1.

[9] L.R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*, Readings in speech recognition **53** (1990), no. 3, 267–296.

[10] L. Raskin, E. Rivlin, and M. Rudzsky, *Using Gaussian Processes for Human Tracking and Action Classification*, Lecture Notes in Computer Science **4841** (2007), 36.

[11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, MIT Press, 2005.

[12] J.M. Wang, D.J. Fleet, and A. Hertzmann, *Gaussian process dynamical models for human motion*, IEEE Transactions on Pattern Analysis and Machine Intelligence (2008).

[13] K. Yamane, Y. Yamaguchi, and Y. Nakamura, *Human motion database with a binary tree and node transition graphs*, Auton. Robots **30** (2011), no. 1, 87–98.

[14] F. Zhou, F. Frade, and J. Hodgins, *Aligned cluster analysis for temporal segmentation of human motion*, IEEE Conference on Automatic Face and Gestures Recognition, 2008.