Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 2012.

Incremental Action Recognition and Generalizing Motion Generation based on Goal-Directed Features

Kathrin Gräve and Sven Behnke

***Abstract*— The ability to recognize human actions is a fundamental problem in many areas of robotics research concerned with human-robot interaction or learning from human demonstration. In this paper, we present a new integrated approach to identifying and recognizing actions in human movement sequences and their reproduction in unknown situations. We propose a set of task-space features to construct probabilistic models of action classes. Based on this representation, we suggest a combined segmentation and classification algorithm which processes data non-greedily using an incremental look-ahead to reliably locate transitions between actions. In a programming by demonstration scenario, our action models afford the generalization and reproduction of learned movements to previously unseen situations. To evaluate the performance of our approach, we consider typical manipulation tasks in a table top setting. In a sequence of human demonstrations, our approach successfully extracts and recognizes actions from different classes and subsequently generalizes them to unknown situations.**

I. Introduction

In order to assist in typical day-to-day tasks, robots need to be able to closely collaborate with humans. This requires the ability to perceive human actions in order to react appropriately. On the other hand, interaction-based methods such as programming by demonstration (PbD) are being actively explored to reduce the programming effort it takes to endow robots with the skills needed for tasks of practical relevance. Programming by demonstration requires the identification of relevant subtasks in movement sequences to reveal inherent task structure and redundancies in the data which can be exploited to limit the computational effort of learning.

In this work, we consider the problem of automatically identifying and extracting actions from a sequence of human movements using an action representation suitable for both, action recognition and reproduction. A central goal of our approach is to extract meaningful actions that can be used as primitives in PbD or other interactive applications (Fig. 1). This is necessary in order to ensure that actions can be reused on their own or be assembled in order to solve more complex tasks. Furthermore, this level of structuring provides a good trade-off between flexibility with respect to various tasks and data efficiency. To this end, a major challenge is how the intuitive notion of a meaningful action can be formalized. Usually, actions are executed in a context and blend into adjacent movements. This makes it difficult to exactly delimit individual primitives.

The authors are with the Autonomous Intelligent Systems group, Department of Computer Science, University of Bonn, Germany. graeve@ais.uni-bonn.de This work was supported by the B-IT Research School.

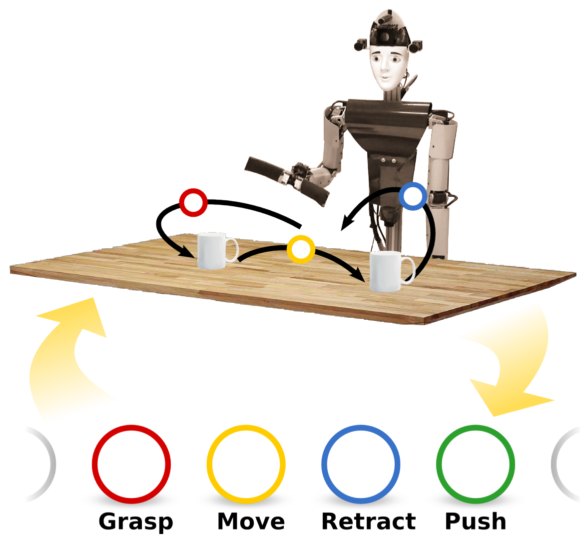

Fig. 1. Illustration of typical actions in a table top scenario extracted from human demonstrations using the presented approach.

Moreover, action classes that are totally different semantically may appear quite similar on a trajectory level and differ mostly by the context they are executed in, although this kind of relevant information is often not available to robots. Instead, data recorded by, e. g., motion capture devices or kinesthetic training usually provides purely geometrical information whose inherent redundancies conceal the essence of the action.

Clearly, segmentation and classification of movement data are two closely related problems whose solutions mutually depend on each other. In this paper, we therefore propose an algorithm which solves both tasks at the same time. Our method relies on Hidden Markov Models (HMM) [1] to encode action classes. This allows us to iteratively delineate meaningful segments corresponding to previously learned actions in unknown movement sequences using a maximum likelihood criterion. We represent actions by generic task space features which allow us to use our model in a generative way to reproduce learned actions and to generalize movements to new situations.

The proposed approach allows to recognize and extract an arbitrary number of known actions from a sequence of demonstrated movements by providing a unified view on segmentation and classification using HMMs. Systematic errors in the choice of segment boundaries due to the sequential processing of input data are avoided using a non-greedy segmentation algorithm. Our approach posesses no free parameters and can readily be applied to a wide range

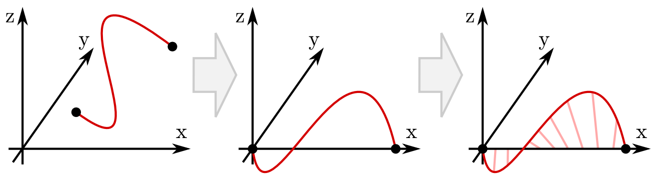

Fig. 2. Illustration of the feature extraction scheme. Left: Original trajectory data. Center: after linear transformation. Right: final features after distance transformation.

of directed movements.

The remainder of this paper is organized as follows: After a brief review of related work in Sec. II, we describe the contributions of this paper in Sec. III, detailing our integrated approach, the spatio-temporal features it uses and our combined segmentation and classification method. We evaluate our approach with respect to segmentation and classification performance and the ability to generalize learned movements using experiments described in Sec. IV.

II. Related Work

Action segmentation, classification, and generation are fundamental problems to the fields of human-robot interaction and programming by demonstration. A lot of research has been devoted to different aspects of these problems but only little work has addressed solving the entire problem in an integrated manner. Offering a solid probabilistic framework, HMMs are a popular tool which has been successfully applied to the segmentation and classification of human movements. Bashir et al. [2], for example, use HMMs to recognize hand gestures from Australian Sign Language. Trajectories are segmented at inflection points into a series of short sub-trajectories and Principal Component Analysis (PCA) is applied to reduce sub-trajectories to feature vectors of a fixed size which are subsequently clustered and encoded in HMMs. Classification is then performed by selecting the HMM which maximizes the observation probability of the data.

Several authors have investigated suitable features for HMM-based classification. For instance, Al Mansur et al. [3] compute dynamics features using an articulated model of the human body. Using torques from four selected joints, their approach allows to classify seven distinct full-body movements. De Schutter et al. [4] developed a coordinate-free representation of 6 DOF rigid body movements which exhibits several invariance properties.

As an alternative to identifying action boundaries with local extrema of features, some researchers considered to use HMMs for both, classification and segmentation. Elmezain et al. [5] proposed a segmentation algorithm based on competitive differential observation probability which measures the likelihood difference between the most likely class given the data and a garbage class [6]. A sign change of this measure is considered as a segment boundary. Kohlmorgen and Lemm [7] proposed an online segmentation algorithm which performes unsupervised segmentation by using HMMs to track the probability density function of the data over time. The segmentation is represented implicitly by the states of the most likely path given the data.

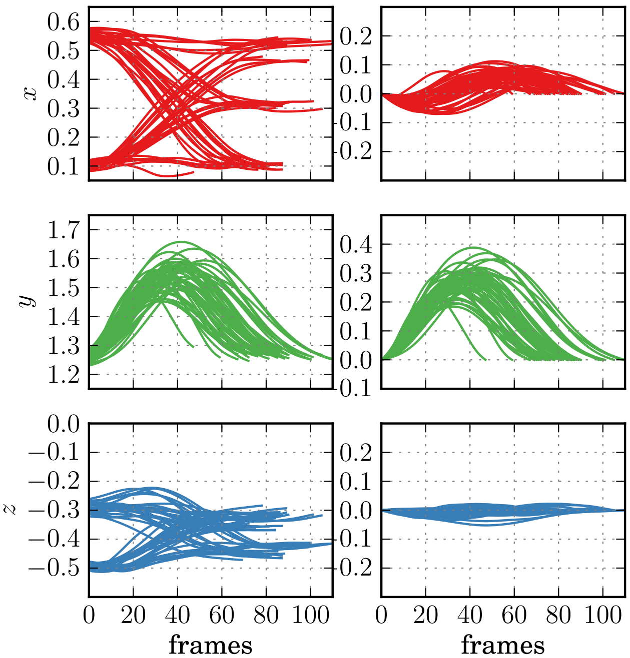

Fig. 3. Illustration of the effect of our linear transformation. Left: original multimodal trajectory data. Right: Transformed unimodal features

In order to reduce the complexity of learning, Li et al. [8] use a two-layered approach inspired by speech recognition. For every relevant object, a set of motion primitives, whose features are also relative to the object, is kept on the bottom layer. The top layer contains transition probabilities among the primitive sets and is responsible for modelling tasks. Sugiura et al. [9] propose to represent actions in an object-dependent coordinate system. Trajectories are encoded using Reference-Point-Dependent Trajectory HMMs which allows for classification and reproduction of presegmented actions. The optimal coordinate system in a maximum likelihood sense is selected during training. Billard et al. [10] consider the problem of generating movements that reproduce the essence of a learned task. They use the variability of joint and task space features to assess their relevance for movement reproduction by a cost function. In a preprocessing step, key points are extracted from the features corresponding to inflexion points. These are encoded in one HMM per dimension. During reproduction, the optimal state sequences are computed and interpolated. In more recent work, Calinon et al. [11] use a combination of Gaussian Mixture Regression and HMM motion models learned from demonstrations to generate movements from a dynamical system. Multiple task constraints are satisfied by combining HMMs learned for various reference frames. In contrast to the HMM-based approaches discussed above, the aim of the present work is to find a holistic approach to classification, segmentation, and generation.

In a similar effort, Kulić et al. [12] recently proposed a

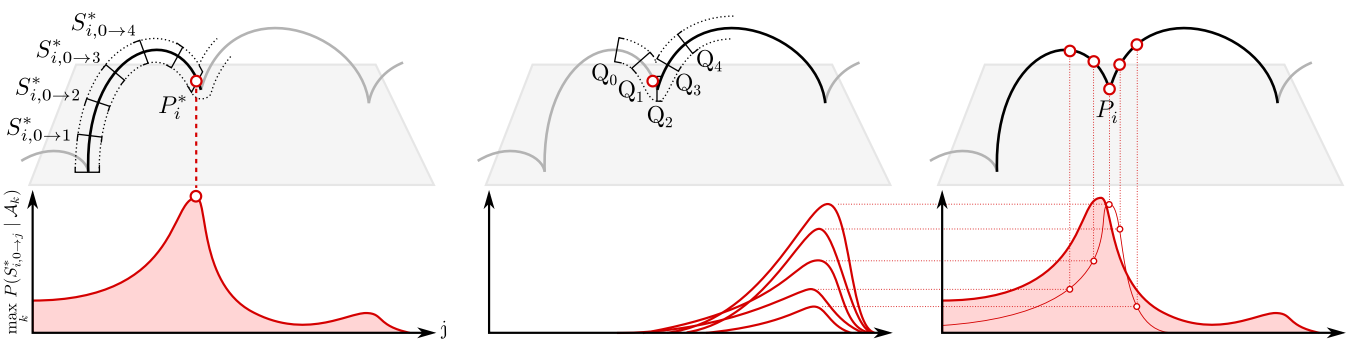

Fig. 4. Schematic overview of our segmentation algorithm. In the top row, a sample movement on a table and relevant segments are depicted. The corresponding likelihoods are shown in the bottom row. In Step 1 (left), the likelihood of candidate segments $S^*_{i,0\to j}$ is computed and the terminal point P^*_i of the most likely candidate is determined. In Step 2 (center), the likelihood of the succeeding segment is computed for origins Q_i in a neighbourhood of P^*_i. In Step 3 (right), for all points Q_i, the likelihoods of the first segment ending at that point and the second segment starting are added. The maximum is considered the segment boundary.

framework that integrates several approaches to incrementally learn human movements. Trajectories are segmented online using an improved variant of Kohlmorgan and Lemm's algorithm [7] and clustered to abstract motion primitives. A HMM is used to encode motion primitives for retrieval and motion reproduction. Sequences of motion primitives can be generated according to a graph that is learned from the observed transition probabilities. While Kulić et al. efficiently combine several models, in our work we strive for a single model suitable for all three tasks.

A recent approach that does not rely on HMMs for the segmentation and classification of movement data has been proposed by Meier et al. [13]. They train Dynamic Movement Primitives (DMP) [14] which allow to generalize actions to arbitrary positions. By reformulating the original framework, the authors obtain an estimate for the goal position and the duration of a partially observed trajectory. Based on these estimates, a segmentation algorithm is devised.

In contrast to these approaches, our work is based on goal-directed spatio-temporal features capturing the characteristics of action classes. Furthermore, instead of processing data in a purely sequential manner, we employ a look-ahead to improve segmentation results.

III. Proposed Method

In this section, we describe our integrated method for action segmentation, classification, and reproduction. The segmentation and classification algorithm is based on a maximum likelihood criterion which takes into account both segments touching a boundary point. It is derived from a single probabilistic model that we use for action segmentation, recognition, and reproduction. This is a major advantage since these tasks are closely related in practical applications. At the core of our approach are Hidden Markov Models which encode spatio-temporal characteristics of actions. The performance of this model relies on a suitable choice of the input features.

A. Spatio-temporal Features

From a modelling perspective, features need to capture a notion of similarity, i. e. there should be little variance among features corresponding to actions of the same category. At the same time, features should be distinctive enough to allow for robust categorization. From an application perspective, action representations should be flexible enough to describe all relevant actions.

Clearly, global Cartesian coordinates do not fulfill the former requirements since a movement is represented by different coordinates if it is executed in a different situation or in a slightly different way. This also reduces the inter-class separability. In this work, we therefore normalize Cartesian trajectory and context information in a preprocessing step before using them as input to an HMM. During normalization, a linear transformation is applied to the input trajectories such that the movement starts at the origin and ends at a point along the first axis. We then consider a straight line between these two points which we sample at regular intervals and use the relative positions of the transformed trajectory with respect to these samples as input to our model. This process is depicted in Fig. 2. As a result, we obtain features which preserve the original movements' characteristics but gain the advantage of invariance with respect to spatial rotations, translations, and different start or end points. This is illustrated in Fig. 3. Furthermore, we include the distance to relevant objects in our feature vector in order to provide context for the classification algorithm. This is motivated by the fact that–depending on the situation–actions may have very different meanings although they are very similar on a trajectory level. In order to recognize these movements correctly, it is necessary to take their context into account.

The feature generation process can easily be inverted in order to generate movements for arbitrary start and end points, given a set of feature vectors.

B. Stochastic Model

To compactly represent learned actions for the segmentation and classification of new movements, we train HMMs on their feature representations. To this end, we assume that a training set of presegmented actions is available. Every action class is represented by a dedicated model whose parameters are incrementally estimated using the Baum-Welch algorithm [1]. The initial state distribution and state

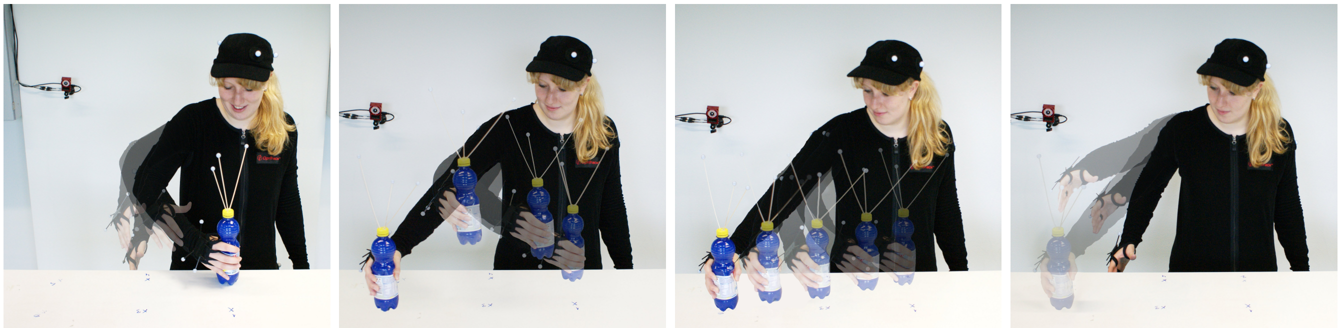

Fig. 5. Actions used throughout our experiments: (left to right) grasping an object, displacing it, pushing it, and retracting the hand to a resting position. For every action, serveral poses are overlaid. The final pose of every action is shown non-transparently.

transition matrices are chosen to constrain the model to traverse all states in linear order, i. e. the only possible state transitions are those to the subsequent or the current state. This choice reflects the sequential structure of the input data. The temporal characteristics of the input data are encoded implicitly in the state transition probabilities of the model.

C. Combined Segmentation and Classification Algorithm

In this section, we present a unified algorithm for probabilistic segmentation and classification of actions.

Since the feature representation of a movement depends on the segmentation, existing solutions to HMM-based segmentation [15] cannot be readily applied. We therefore propose an iterative approach to delineating an action in a continuous movement sequence. This process is depicted in Fig. 4. Instead of taking into account only the frames processed up to a potential segment boundary, the algorithm performs a look-ahead in order to assess the probability of another segment starting. The boundary is determined by maximizing the likelihood of both segments. The algorithm takes the following steps to delineate an action:

Step 1:

In this step, a candidate P_i^* for the intersection point P_i separating the current segment S_i from its successor S_{i+1} is computed. The algorithm starts at the beginning of S_i with an empty preliminary segment $S^*_{i,0\to 0}$ and incrementally appends trajectory points, forming preliminary segments $S^*_{i,0\to 1}, \ldots, S^*_{i,0\to n}$. In every iteration, features for $S^*_{i,0\to j}$ are computed using the scheme outlined in Sec. III-A. Using these features, the likelihood

$$P(S^*_{i,0\to j} \mid \mathcal{A}_k) \qquad \forall j,k \tag{1}$$

of the segments $S^*_{i,0\to j}$ with respect to the action models $\mathcal{A}_k$ is computed using the Viterbi algorithm [1] and the most likely model $\mathcal{A}_j^{best}$ is recorded for every $S^*_{i,0\to j}$ along with its likelihood. Among all $S^*_{i,0\to j}$, the one with the maximum likelihood is determined and its terminal point is returned as the potential intersection point P_i^*.

Step 2: The choice of an intersection point not only delineates the action preceeding it but also defines the start of the succeeding action. To take this influence into account, the potential intersection point P_i^* is improved by considering points Q_l in a local neighborhood as potential origins of a succeeding action S_{i+1}. By executing Step 1 starting from these potential origins, a likelihood according to Eq. (1) of the succeeding action starting at that point and a corresponding terminal point are computed for each of them.

Step 3: For each point evaluated in Step 2, a combined likelihood is computed from the results of Step 1 and Step 2 which takes into account the likelihood of the point being the terminal point of segment S_i as well as the likelihood of it being the origin of segment S_{i+1}. The point P_i maximizing this likelihood is determined according to

$$\begin{aligned} p = \arg\max_l \{ & \log P(S^*_{i,0\to l} \quad | \mathcal{A}_l^{best}) \\ & + \log P(S^*_{i+1,l\to m} | \mathcal{A}_m^{best}) \}, \end{aligned} \tag{2}$$

where $\mathcal{A}_m^{best}$ is the most likely model determined for $S^*_{i+1,l\to m}$ in Step 2. The segment S_i is classified according to the likelihoods of the model $\mathcal{A}_l^{best}$. In order to determine the next intersection point, the procedure is repeated, reusing the terminal point of $S^*_{i+1,l\to m}$ found in Step 2 of the first run as result P^*_{i+1} of Step 1 in the second run.

D. Motion Generation

Our choice of a probabilistic model and the features we encode therein allows us to reproduce the learned actions and to generalize them to new situations. To generate a movement from a learned class between given start and terminal points, we first recover a sequence of key points from the model. We then obtain a set of features through interpolation from which we compute the desired trajectory.

Encoding features in HMMs allows us to employ a straightforward scheme in order to obtain a sequence of feature vectors for a given action which is largely inspired by [16]. Due to our model topology, the hidden states of our model are traversed sequentially. Following [16], we use the mean values of the hidden states' Gaussian observation models as keypoints of the desired feature sequence. In contrast to their approach, we use Gaussian Process Regression (GPR) [17] to interpolate the features between the keypoints. GPR implicitly models the statistics of the data and provides a trade off between a close fit to the data and a smooth approximation. Once a sequence of features is determined, the final trajectory from an arbitrary origin to a user-defined end point can easily be computed by

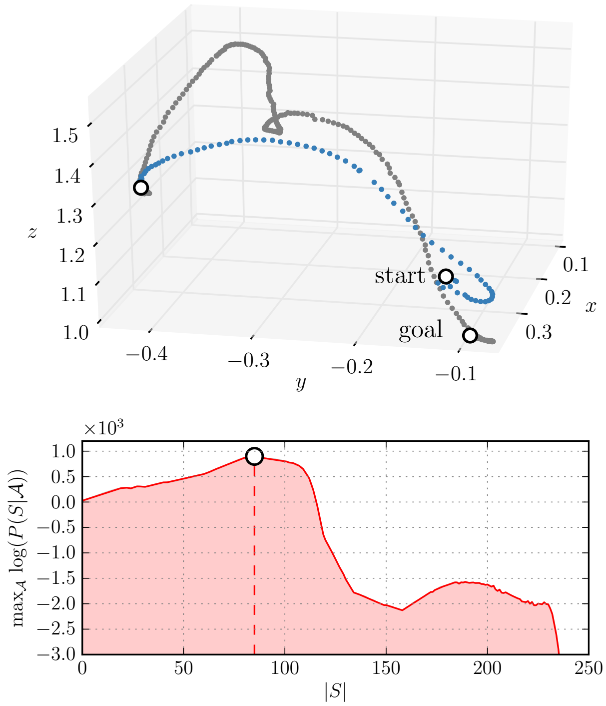

Fig. 6. Sample trajectory where the segments have been correctly identified by our approach (top). The boundary delimiting the first segment has been identified by maximizing the log likelihood of candidate points depicted in the bottom graph.

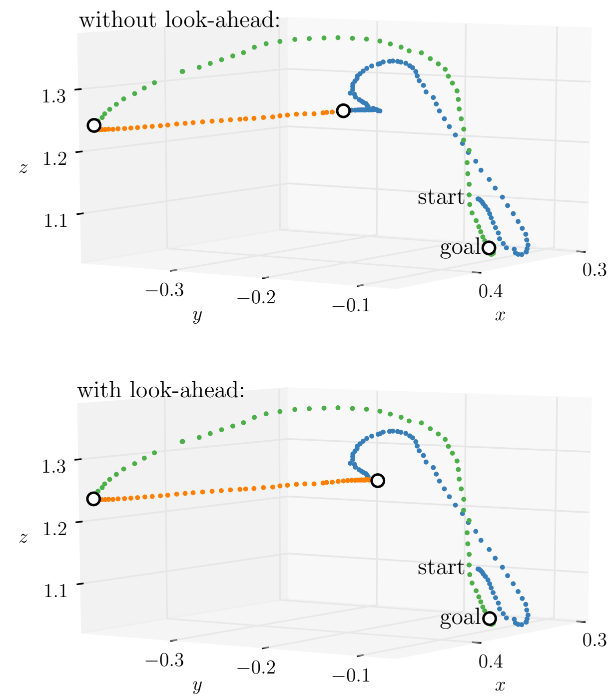

Fig. 7. Top: Segmentation which does not take segments following a boundary point into account. This leads to a suboptimal result as the grasping movement (blue) streches into the push movement (orange). Bottom: Improved segmentation obtained by also considering the successor.

reversing the procedure described in Sec. III-A using the desired origin and terminal points as inputs to the procedure. Since the features themselves are independent of these points, and only encode the shape of the trajectory, the procedure easily generates movements even for previously unknown situations.

IV. Experiments

We validate our approach in a series of experiments. As an exemplary task, we choose manipulation movements in a table top scenario, which is relevant to many real-world applications for service robots. As shown in Fig. 5, we consider four different classes of manipulation actions, namely grasping an object, displacing it, pushing it to a different position without lifting it, and retracting the hand to a convenient resting position. These primitive movements can be chained in order to solve complex tasks.

A. Experimental Setup

The 3D trajectory data for our experiments was recorded using an optical motion capture rig at a rate of 100 Hz. This provides us with the Cartesian positions of a human's hand and relevant objects. We recorded several demonstrations for every action class to train the models. Their end points were chosen from six predefined points located in front of the demonstrator, to her left, and to her right at a distance of 25 cm and 40 cm, respectively. For every location, we separately demonstrated ten times how to grasp an object and how to retract the hand. Furthermore, we gave five demonstrations of a displacement and push movement from two points to all others. The different end points led to variations in the movements' shape.

To reduce the effects of occlusions and noise, we applied temporal resampling and spatial smoothing to the motion capture data before computing the features for training. The reference objects we use throughout our experiments to provide context information to the feature calculation are the table top and the object we manipulate. Therefore, the input to our model is a five dimensional vector containing the 3D features computed from the hand trajectory, the vertical distance of the end effector to the table's surface, and the Euclidean distance to the object.

B. Combined Segmentation and Classification

To validate our approach and demonstrate its performance, we recorded another set of sequences containing different numbers of actions. Since our recognition of actions is robust with respect to noise in the input data, we do not need to preprocess the recordings.

In our approach, we find action boundaries by maximizing the likelihood of the input data with respect to the learned action classes. Fig. 6 shows a sample trajectory and the boundary of the first segment found by our algorithm, as well as the development of the log likelihood during the search process using an incrementally growing window. The likelihood increases almost monotonically, with the exception of a few deviations caused by outliers in the input data. At the terminal point of the first segment, the likelihood exhibits a

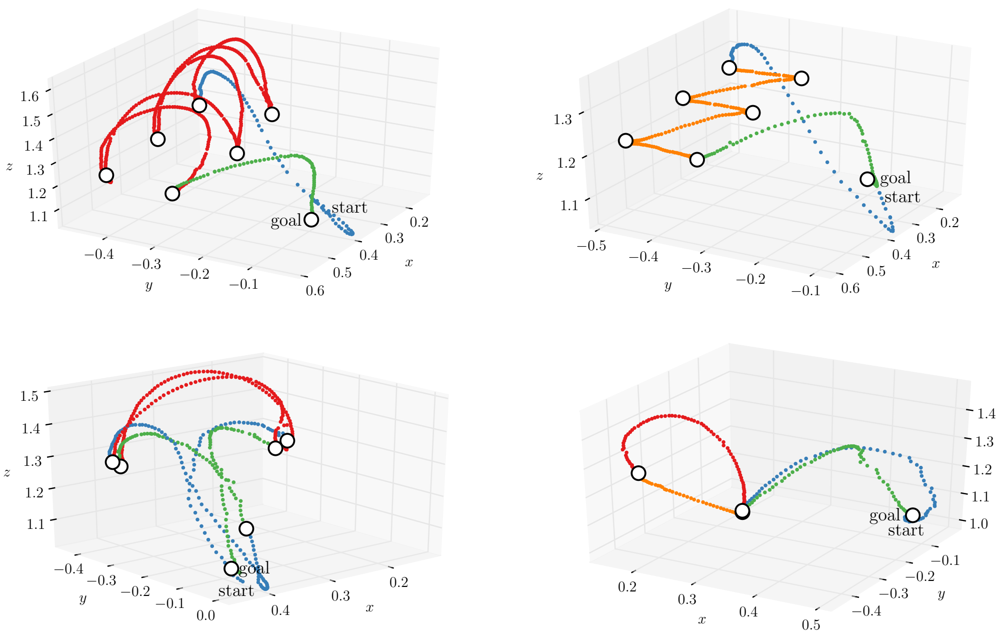

Fig. 8. Examples of complex movement sequences successfully segmented and classified by our algorithm. Actions are represented by colors: Grasping (blue), displacing an object (red), pushing an object (orange), and retracting the hand to a resting position (green). The segment boundaries are highlighted using bullets.

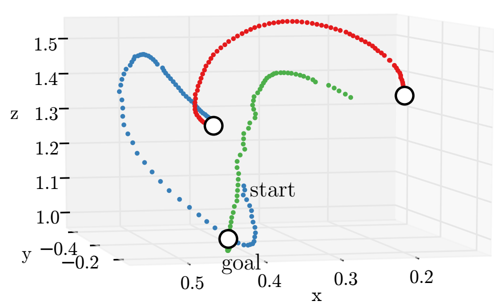

Fig. 9. Performance of our approach on distorted data. The example movement is correctly recognized in spite of a large gap in the data between the displacement movement (red) and retraction movement (green).

pronounced global maximum. By further increasing the window size, we find another local maximum of the likelihood corresponding to the end of the second segment. This is due to the shape of the combined movement which at a coarse level resembles a grasping movement in the feature space.

Although the segment boundaries determined this way are often correct, there are some cases where considering only the segment preceeding a boundary is not sufficient. Fig. 7 depicts the difference between choosing a segment boundary considering only the preceeding segment and the boundary obtained by also taking into account the following segment. Whereas the first choice leads to a segment extending into its successor, taking into account the likelihood of the latter yields the correct boundary.

To quantitatively assess the performance of our approach, we recorded 107 sequences containing three successive actions each. At the beginning of every sequence, we grasped an object on a table. We then either lifted or pushed the object to a different location and finally retracted the hand. The movements were executed in the same area that was used to train the models–we did not, however, limit our experiments to the trained positions but also recorded actions at intermediate places. We applied our algorithm to segment and classify the recorded sequences and counted the number of erroneous segmentations and misclassifications. Among the 321 recorded actions, 13 were incorrectly delineated by our algorithm. This corresponds to an accuracy of 96%. The errors were mostly caused by particularly sweeping grasping movements that were split into two separate actions. Subsequent segments were not influenced by these mistakes. Our classification alorithm achieved an accuracy of 100% on the correct segments and did not misclassify a single action.

The previous experiment demonstrated the performance of our approach on exemplary sequences of three segments. The ability of our approach to segment sequences composed of a larger number of actions is demonstrated in Fig. 8. All shown sequences start with a grasping movement followed by a series of displacement and push actions on the grasped object. Finally, the hand is retracted to a convenient resting position. The examples are chosen representatively and demonstrate that all individual actions are correctly delineated and classified.

Our approach copes well with long movement sequences and easily handles difficulties such as noise, outliers and gaps in the input data. This is demonstrated in Fig. 9 which contains examples of degraded trajectories that are segmented and classified correctly.

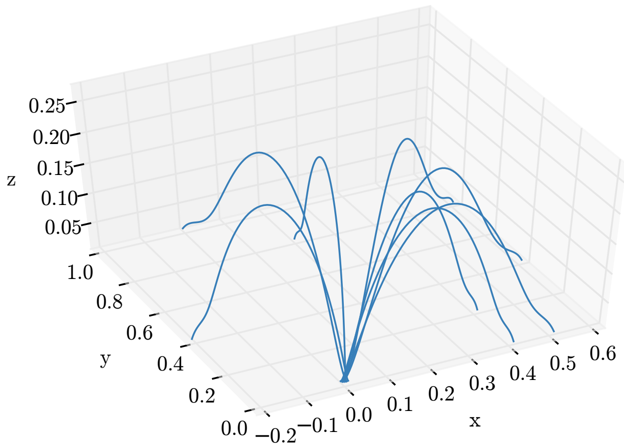

Fig. 10. Generated displacement movements to various positions on a table.

C. Movement Reproduction and Generalization

The aim of movement reproduction is to generate trajectories for arbitrary situations which exhibit the characteristic features of the learned action while abstracting from the specifics of individual trained movements. To this end, we propose to apply GPR to extract smooth features from the state sequence encoded in HMMs.

The generalization abilities of our approach are demonstrated in Fig. 10. Several displacement actions are reproduced between various positions on a table. Only the desired starting and ending points in task space are used as input to the algorithm. The shape of the generated movements is solely determined by the learned models. The plots show that the characteristic shape of the action is preserved across the various reproductions. To execute the generated trajectory on a robot, it is important to verify the kinematic feasibility.

V. Conclusions

In this paper, we proposed an incremental algorithm for probabilistic action recognition and generation based on HMMs. A linear transformation of features from task space and a representation relative to the endpoints of movements leads to a low intra-class variance which allows for accurate segmentation, classification, and reproduction of actions using a common model. To obtain meaningful segmentation results, we developed an algorithm that takes into account the likelihood of actions on both sides of segment boundaries. Our experiments show that segmentation results are improved by this strategy. We demonstrated that the approach is capable of correctly segmenting sequences of several actions. On a sample set of 321 actions, we achieved an accuracy of 96% for segmentation and 100% for classification of correctly segmented actions. By applying GPR to key points extracted from our model, our approach is able to reproduce learned actions retaining the essence of the original movement. The choice of our features also allows to generalize reproductions to unknown situations. In future work, we would like to extend our approach to handle multiple objects and to add the ability to recognize objects relevant to the task. Another way to extend our method is to investigate the use of recognized actions as training samples to improve the performance of the underlying models.

References

[1] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, 1989.

[2] F. I. Bashir, A. A. Khokhar, and D. Schonfeld. Object Trajectory-based Activity Classification and Recognition Using Hidden Markov Models. *IEEE Transactions on Image Processing*, 2007.

[3] A. Mansur, Y. Makihara, and Y. Yagi. Action Recognition using Dynamics Features. In *IEEE International Conference on Robotics and Automation, (ICRA), Shanghai, China*, 2011.

[4] J. De Schutter, E. Di Letto, J.F.M. De Schutter, R. Matthysen, T. Benoit, and T. De Laet. Recognition of 6 DOF Rigid Body Motion Trajectories using a Coordinate-Free Representation. In *IEEE International Conference on Robotics and Automation, (ICRA), Shanghai, China*, 2011.

[5] M. Elmezain, A. Al-Hamadi, and B. Michaelis. Hand Trajectory-based Gesture Spotting and Recognition using HMM. In *International Conference on Image Processing, (ICIP), Cairo, Egypt*, 2009.

[6] H. Yang, A. Park, and S. Lee. Gesture Spotting and Recognition for Human-Robot Interaction. *IEEE Transactions on Robotics*, 23(2), 2007.

[7] Jens Kohlmorgen and Steven Lemm. A dynamic HMM for on-line segmentation of sequential data. In *Advances in Neural Information Processing Systems 14*, pages 793–800. MIT Press, 2002.

[8] Z. Li, S. Wachsmuth, J. Fritsch, and G. Sagerer. Manipulative Action recognition for human-robot interaction. In *Vision Systems: Segmentation and Pattern Recognition*, chapter 8. I-Tech Education and Publishing, 2007.

[9] K. Sugiura, N. Iwahashi, H. Kashioka, and S. Nakamura. Learning, Generation and Recognition of Motions by Reference-Point-Dependent Probabilistic Models. *Advanced Robotics*, 25(6-7), 2011.

[10] A. Billard, S. Calinon, and F. Guenter. Discriminative and Adaptive Imitation in Uni-Manual and Bi-Manual Tasks. *Robotics and Autonomous Systems*, 54(5), 2006.

[11] S. Calinon, F. Dhalluin, D. Caldwell, and A. Billard. Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework. In *IEEE International Conference on Humanoid Robots*, pages 582–588, 2009.

[12] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura. Incremental learning of full body motion primitives and their sequencing through human motion observation. *The International Journal of Robotics Research*, 31(3):330–345, 2012.

[13] F. Meier, E. Theodorou, F. Stulp, and S. Schaal. Movement Segmentation using a Primitive Library. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS), San Francisco, USA*, 2011.

[14] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots. 2002.

[15] G. A. Fink. *Markov Models for Pattern Recognition, From Theory to Applications*. Springer, Berlin, Heidelberg, 2008.

[16] F. Guenter, S. Calinon, and A. Billard. Goal-Directed Imitation in a Humanoid Robot. In *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[17] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.