Probabilistic semantic models for manipulation action representation and extraction

Rok Vuga[a,*], Eren Erdal Aksoy[b], Florentin Wörgötter[b], Aleš Ude[a]

[a]*Humanoid and Cognitive Robotics Lab, Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia*
[b] *Georg-August-Universität Göttingen, BCCN, Department for Computational Neuroscience, Inst. Physics-3, Friedrich-Hund Platz 1, D-37077 Göttingen, Germany*

Abstract

In this paper we present a hierarchical framework for representation of manipulation actions and its applicability to the problem of top down action extraction from observation. The framework consists of novel probabilistic semantic models, which encode contact relations as probability distributions over the action phase. The models are action descriptive and can be used to provide probabilistic similarity scores for newly observed action sequences. The lower level of the representation consists of parametric hidden Markov models, which encode trajectory information.

Keywords: probabilistic semantic models, semantic event chains, manipulation action primitives, action extraction, parametric hidden Markov models

1. Introduction

An autonomous learning robot should be able to extract knowledge by visually observing a human performing a desired task. Ideally, the robots should learn in a manner similar to humans, who are capable of lifelong refinement of their skills. Therefore we wish that the robot would autonomously update its internal action models as it comes across new demonstrations. In this paper we present a novel framework for representation of manipulation actions, which is useful for locating known actions in longer observations.

[*] Corresponding author. Phone: +386 1 477 3463
Email address: rok.vuga@ijs.si (Rok Vuga)

Similar approaches in the literature, which use top down search for actions, typically deal with the case where all of the actions contained in the sequence are known. The algorithm only finds segmentation points between these actions. However, a robot which continuously monitors its environment would most of the time observe unrelated phenomena, with occasionally useful demonstration of a known action. Such top-down search for segmentation points using hidden Markov models as trajectory models and dynamic programming was presented in [1]. In a similar approach, Hoai et al. [2] use multi-class support vector machines. There has also been some research in speech processing, where the problem is known as keyword spotting [3]. These approaches mostly use hidden Markov models. On the other hand, Kang and Ikeuchi [4] dealt with temporal segmentation of manipulation tasks. They present a measure for detecting segmentation points based on the speed of demonstrator's hand trajectory and type of his grasp. They do not use action models, but instead segment the task based on low level trajectory properties.

Herein we present a two-layer action representation, where the upper, semantic layer, consists of novel probabilistic semantic models which encode touch relations. The lower, trajectory layer, consists of parametric hidden Markov models. We use the proposed semantic action models in order to provide initial segmentation hypotheses. Search in this constrained space is then performed using low level trajectory models.

At least in manipulation tasks, relations between objects and hands involved in the action have been considered to extract higher level information. A large body of works deals with so called topological contact states between tasks, which characterize contact relations between manipulated objects. Different contact state representations differ in the way contacts are defined [5]. One of the early examples is the work of Lozano-Perez [6], which deals with spatial planning of manipulations for polygonal and polyhedral objects, representing contacts in terms of vertex-face and edge-edge contacts. Ikeuchi and Suehiro [7] present configuration changes between objects before and after manipulation as high level assembly descriptions, which can be extracted from observation. Xiao [8] introduces the notion of principal contacts, which are elementary contacts between objects, such that seven different contacts are possible between a pair of 3-D objects. In [9] they present a method to construct contact formation graphs as representations of the elementary contacts. Recently, Aksoy et al. [10] proposed the Semantic Event Chain (SEC) framework, which characterizes manipulation actions by the following two spatial relations: *touching* and *not touching.* They provide a general computer vision system to extract these relations and show that their framework is sufficient to represent all possible elementary single handed manipulations [11]. Yang et al. [12] developed an algorithm to monitor changes in object appearance and topological structure, which is used to infer consequences of actions and recognition of manipulations. Many works also explore object-action affordances [13, 14, 15, 16]. These approaches are more object-centric, as they deal with determining the roles of the objects in manipulation tasks and are less concerned with the problem of action execution.

In this paper we build upon the notion of semantic events, as presented in [10]. We construct action models not by deterministically assigning contact relations to image segment pairs, but by probabilistically modeling frequencies of semantic events. The proposed methodology is robust against noise, which can cause spurious relations between objects to arise. We start with a brief explanation of the SEC framework (Section 1.1), followed by issues, typically encountered when attempting to extract symbolic representations from sensory data (Section 1.2). Section 2 gives a detailed explanation of the proposed probabilistic semantic models. Experiments using the proposed approach are presented in Section 3. Compared to some other works in the literature [7, 8], which focus on accurate analysis of object relations, our approach relies on statistics of the observed contact relations and is therefore very suitable for applications that involve processing of real-world vision data.

1.1. Semantic object relations

Semantic event chains, developed by Aksoy et al. [10], introduced the idea of annotating relations between objects at decisive time points. The aim is to construct a symbolic task description. Inspired by language development [17], semantic relations are defined by object-object relationships. For each pair of objects found in the scene, a semantic relation can be defined as *touching*, *not touching*, or either of the two *not present* in the scene. A semantic event is defined to occur whenever a change in semantic relations happens in the scene. This can be caused by one of the following situations: a new pair of objects starting to touch, a pair ceasing to touch, a new object entering the scene, or an object leaving the scene. A semantic event chain is then constructed as a temporal record of changes in semantic relations for all pairs of objects in the scene.

1.2. Challenges for working with real-world data

As already noted in [10, 18], it is naive to expect that the extraction of object relations in complex scenes by general purpose computer vision would be prefect. To compute relations between objects, objects must first be segmented from the background. Due to the fact that in a learning scenario, completely unknown objects may be encountered, model free object segmentation algorithms must be used. Since model free object segmentation is an ill-posed problem, typical algorithms rely on heuristic assumptions of "what an object is". Examples include assuming color or shape properties [19] or rigid body motion principles [20, 21]. This way, a rough correspondence between extracted image segments and actual objects can be obtained. However, detecting and segmenting unknown objects without any errors in the presence of unpredictable motion, occlusions, even changes in objects themselves (e. g. an object being broken into pieces), is not realistic in the near future.

Hence we are limited to noisy sensory data to extract information from the real world. Needless to say, noise accompanying the measurements affects the final output of the system. For example, in the process of spatial contact detection some erroneous object relations can be detected for neighbouring objects. Any symbolic representation which relies on such spatial object relations will be prone to errors such as oversegmentation of manipulation sequences. Furthermore, real world scenes often contain clutter - objects that are not an essential part of the task and just happen to be in the robot's field of view. Methods which deterministically construct symbolic representation from observation may treat any semantic events involving these objects as essential parts of the action.

For example, consider a semantic event chain for a simple task of pouring some milk into a bowl. A semantic event chain for this task should have few entries: the hand touches the milk box, the milk appears in the bowl, the hand puts the box back on the table and stops touching it. Figure 1 illustrates what typically happens when we extract the chain from a real-life demonstration. Since we used computer vision as a sensory system, the chain was constructed based on relations between image segments, which are just approximations of true objects and as such the chain contains many items which do not have a basis in physical object relations. As you can see, the obtained result is far from optimal.

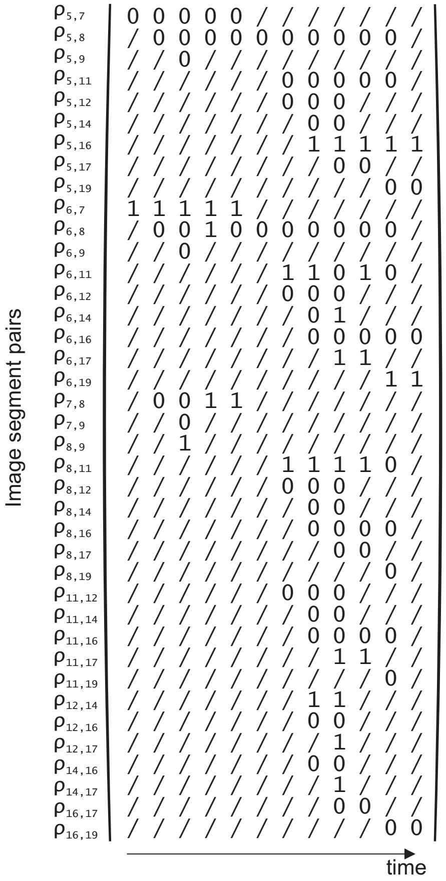

Image segment pairs												
$\rho_{5,7}$	0	0	0	0	0	/	/	/	/	/	/	/
$\rho_{5,8}$	/	0	0	0	0	0	0	0	0	0	0	/
$\rho_{5,9}$	/	/	0	/	/	/	/	/	/	/	/	/
$\rho_{5,11}$	/	/	/	/	/	/	0	0	0	0	0	/
$\rho_{5,12}$	/	/	/	/	/	/	0	0	0	/	/	/
$\rho_{5,14}$	/	/	/	/	/	/	/	0	0	/	/	/
$\rho_{5,16}$	/	/	/	/	/	/	/	1	1	1	1	1
$\rho_{5,17}$	/	/	/	/	/	/	/	/	0	0	/	/
$\rho_{5,19}$	/	/	/	/	/	/	/	/	/	/	0	0
$\rho_{6,7}$	1	1	1	1	1	/	/	/	/	/	/	/
$\rho_{6,8}$	/	0	0	1	0	0	0	0	0	0	0	/
$\rho_{6,9}$	/	/	0	/	/	/	/	/	/	/	/	/
$\rho_{6,11}$	/	/	/	/	/	/	1	1	0	1	0	/
$\rho_{6,12}$	/	/	/	/	/	/	0	0	0	/	/	/
$\rho_{6,14}$	/	/	/	/	/	/	/	0	1	/	/	/
$\rho_{6,16}$	/	/	/	/	/	/	/	0	0	0	0	0
$\rho_{6,17}$	/	/	/	/	/	/	/	/	1	1	/	/
$\rho_{6,19}$	/	/	/	/	/	/	/	/	/	/	1	1
$\rho_{7,8}$	/	0	0	1	1	/	/	/	/	/	/	/
$\rho_{7,9}$	/	/	0	/	/	/	/	/	/	/	/	/
$\rho_{8,9}$	/	/	1	/	/	/	/	/	/	/	/	/
$\rho_{8,11}$	/	/	/	/	/	/	1	1	1	1	0	/
$\rho_{8,12}$	/	/	/	/	/	/	0	0	0	/	/	/
$\rho_{8,14}$	/	/	/	/	/	/	/	0	0	/	/	/
$\rho_{8,16}$	/	/	/	/	/	/	/	0	0	0	0	/
$\rho_{8,17}$	/	/	/	/	/	/	/	/	0	0	/	/
$\rho_{8,19}$	/	/	/	/	/	/	/	/	/	/	0	/
$\rho_{11,12}$	/	/	/	/	/	/	0	0	0	/	/	/
$\rho_{11,14}$	/	/	/	/	/	/	/	0	0	/	/	/
$\rho_{11,16}$	/	/	/	/	/	/	/	0	0	0	0	/
$\rho_{11,17}$	/	/	/	/	/	/	/	/	1	1	/	/
$\rho_{11,19}$	/	/	/	/	/	/	/	/	/	/	0	/
$\rho_{12,14}$	/	/	/	/	/	/	/	1	1	/	/	/
$\rho_{12,16}$	/	/	/	/	/	/	/	0	0	/	/	/
$\rho_{12,17}$	/	/	/	/	/	/	/	/	1	/	/	/
$\rho_{14,16}$	/	/	/	/	/	/	/	0	0	/	/	/
$\rho_{14,17}$	/	/	/	/	/	/	/	/	1	/	/	/
$\rho_{16,17}$	/	/	/	/	/	/	/	/	0	0	/	/
$\rho_{16,19}$	/	/	/	/	/	/	/	/	/	/	0	0

Figure 1: The semantic event chain extracted from a recording of a milk pouring task. The entries in the table present semantic relations between given pairs of labeled image segments. Zero denotes the pair not touching, one denotes touching, and “/” denotes either of the two not present. Each consecutive column corresponds to each image still from Figure 7. The chain is not a good representation of the milk pouring task as it is over-segmented. First, due to unreliable contact detection from vision, many contacts were detected multiple times, resulting in too many columns. Second, since no object models are available, the same objects were detected multiple times, which resulted in too many rows.

1.3. Hierarchical approach and lower level methods

In our previous work [17] we presented an approach where actions are represented at two levels. The semantic level, which encodes object relations, and descriptor level, which holds lower level information about execution of the action. With such approach we can model the fact that some actions are semantically similar, but are importantly different in terms of low level descriptors. This simplifies evaluation of unknown sequences, as comparison of lower level features is needed only in cases where a definite match could not be found at the semantic level.

In this paper we follow the same scheme - we use our novel probabilistic models on the semantic level and use trajectory information to represent the lower level.

Representation and recognition of activities based on trajectory-level features has long history. Probably the most popular approach are hidden Markov models (HMM, [22]) and their many derivatives. They are a general framework for statistical modeling of patterns, with successful applications in speech [23], handwriting [24], as well as gesture recognition [25]. They have also been utilized for human activity recognition, for example detecting running or jumping from video feeds [26] [27] [28]. In our work we deal with human manipulation actions, which are short and discrete. Compared to gesture recognition problem, differences in execution trajectories between different manipulations are much more subtle.

Herzog et al. [29] examined modeling of human and robot trajectories using parametric hidden Markov models (PHMM, [30]). PHMMs can capture inter-class variations between examples and as such ideal for representing manipulation trajectories. Details regarding training and usage can be found in Section 2.4.

2. Our approach

We propose to model actions on two levels. At the semantic level we, for the reasons presented in Chapter 1.2, avoid construction of higher level symbols and focus on the frequency of events, typically defining the symbols. Following the SEC nomenclature, we concentrate on semantic events - changes of semantic relations extracted from an image stream. We do not consider concrete relations between image segment pairs. Instead, we define models of action semantics as distributions of semantic events over action duration. The models are learned in a supervised way from a set of training demonstrations. From the same training set, we also build models using execution trajectory data, which serves as lower level data. This allows the system to evaluate previously unseen sequences in a hierarchical way. First the pattern of occurrence of semantic events is compared to the known distributions. Semantically similar actions are further distinguished by comparing trajectories to trajectory models. We present construction of the semantic models first; details on low level trajectory modeling can be found in Section 2.4.

With the proposed approach we model manipulation actions. By the term "action" we consider a part of a manipulation process, which is atomic in a sense that it is captured by a single semantic model. Since the presented method of training the models is supervised, the action is defined by the data in the training set. There is no limit on how "primitive" an action needs to be with respect to the manipulation process; depending on the user's need, the semantic model could be trained to either model a small part of the manipulation, or the whole assembly from start to end. In the experiments in this paper we portray with former case. We use the term "task" to denote the whole manipulation process (e.g. task of opening the box) and the term "action" to denote an integral part of the task (e.g. reaching for the lid).

Construction of the proposed semantic models thus starts with system observing several demonstrations of a particular action. Semantic model of the action is then obtained by discretizing phase space and calculating probability of observing semantic events for all phase intervals. The most important insight for our work is that while many of detected events are due to noise, the ones which relate to real changes in the scene will be detected consistently throughout demonstrations at similar phases. Provided with a sufficiently large training set, their occurrence can be probabilistically modeled.

From each demonstration the semantic events are extracted in the form of a so called semantic event sequence. Denoted here by E, semantic event sequence is an indicator function over the time domain. Defined at every time step t, it takes a discrete value, denoting whether a semantic event was recorded at that particular time step or not. Hereinafter, we will refer to it as taking value 1 for "recorded" and 0 for "not recorded". Formally, this can be written as binary function

$$E : [0, t] \mapsto \{0, 1\}. \tag{1}$$

Therefore, whenever a new object segment is recognized in the scene, disappears from the scene, or a pair starts to touch or overlap or stops doing so, E takes value 1. This includes cases where an image segment was recognized as a new object after occlusion or rotation, which caused that what used to be backside is now visible, when two objects are detected as touching due to noisy depth information, and other outcomes causing noise in our data.

2.1. Phase mapping

When a human performs an action, the speed of execution is not constant over different performances of the same action. We therefore first normalize the timing of the recorded data:

$$\Phi : t \mapsto \phi$$

$$\phi(t) = \frac{t - t_0}{t_{end} - t_0}, \quad t_0 \leq t \leq t_{end}. \tag{2}$$

Variable ϕ is called phase and starts at 0 and finishes with 1. This way we achieve independence of our data from time scale, which enables us to compare examples of different durations.

After the transformation, we obtain a semantic event sequence defined over the domain of ϕ, such that $E'(\phi) = 1$ for each phase ϕ where a semantic event occurs and zero for all other phases:

$$E' : [0, 1] \mapsto \{0, 1\} \tag{3}$$

$$E' = E\left(\Phi^{-1}(\phi)\right) \tag{4}$$

Note that there exist other possibilities for defining the phase mapping. For example, dynamic time warping [31] could be used for better temporal alignment. See Section 3.3 for discussion on time warping and normalization of time.

2.2. Training

Provided with a large enough set of training data, consisting of N_{ex} semantic event sequences $\{\mathcal{E}_1, \ldots, \mathcal{E}_{N_{ex}}\}$, where $\mathcal{E}_n$ is a recording of E' during n-th execution of the given action, we can find the frequencies of event occurrences over the action phase. We start by dividing the phase into N_s discrete bins

$$S = \{s_1, s_2, \ldots s_{N_s}\}. \tag{5}$$

Each consecutive state s_i models successive parts of the phase interval $[0,1]$, which are determined as

$$\Psi_i = \left[\frac{i-1}{N_s}, \frac{i}{N_s}\right), \quad i = 1 \dots N_s - 1, \tag{6}$$

$$\Psi_i = \left[\frac{i-1}{N_s}, \frac{i}{N_s}\right], \quad i = N_s. \tag{7}$$

We continue by defining the number of observed events in a given state during training, denoted here by N_i^{tr}

$$O_i^n = \begin{cases} 1, & \exists \phi \text{, such that } \phi \in \Psi_i \wedge \mathcal{E}_n(\phi) = 1 \\ 0, & \text{otherwise} \end{cases}, \tag{8}$$

$$N_i^{tr} = \sum_{n=1}^{N_{ex}} O_i^n. \tag{9}$$

Simply, if one or more events occurred during phase interval Ψ_i in a particular training sequence $\mathcal{E}_n$, O_i^n will be one. If no semantic events were detected, O_i^n will be zero. N_i^{tr} is a summation over the complete training set and provides us with the number of training examples which had at least one event occur during phase interval Ψ_i.

We can now calculate the parameters of our model. It is a probabilistic model, consisting of several states, each of which corresponds to a phase interval. The phase variable takes care of progressing through the states; starting in state s_1 at the beginning of an action and ending in the final state s_{N_s} at the end of the action. Upon visiting a state s_i, the process outputs a discrete random observation variable O_i, which can take two possible values, corresponding to whether a semantic event has been observed or not. The output probabilities for state s_i are given as

$$P(O_i = 1) = \frac{N_i^{tr}}{N_{ex}}, \tag{10}$$

$$P(O_i = 0) = 1 - P(O_i = 1). \tag{11}$$

Probability of observing an event during a certain phase interval is proportional to the number of training examples which did produce an event during that part of the action. This way, states with high probability $P(O_i = 1)$ signify that for the given action, semantic events are very likely to occur during the corresponding phase intervals. Likewise, it is less likely to observe semantic events in phase intervals with high probability $P(O_i = 0)$.

The output probability defined above is a measure of how probable the observation of an event is for a particular action. If an event occurred during a certain phase of the action every time the action was executed, it is unlikely to be just a result of a noisy observation. The opposite holds for states with unobserved events; if an event was observed during training just in a few of the examples, then it is most probably not an integral part of the action execution, but instead a result of sensory noise.

We write the calculated probabilities into the observation matrix $\mathbf{B}$, with dimensions $2 \times N_s$, such that

$$\begin{aligned} \mathbf{B}_{1,i} &= P(O_{s_i} = 0) \\ \mathbf{B}_{2,i} &= P(O_{s_i} = 1). \end{aligned} \tag{12}$$

Thus our semantic model is fully defined by

$$\lambda = \{N_s, \mathbf{B}\}. \tag{13}$$

See also Figure 2 for illustration of the proposed framework.

2.3. Action recognition

We can use the learned models to evaluate similarity scores for an observed action. Suppose we observe a demonstrator execute a manipulation action and record a semantic event sequence $\mathcal{E}^*$. We want to calculate the likelihood of this signal belonging to one of the trained action models.

After normalizing the timing using Eq. (4), we check for each of the N_s phase intervals whether there were any events or not. That is:

$$O_i^* = \begin{cases} 1, & \exists\phi, \text{ such that } \phi \in \Psi_i \wedge \mathcal{E}^*(\phi) = 1 \\ 0, & \text{otherwise} \end{cases}. \tag{14}$$

Similarity of the unknown action versus some model λ is then obtained by computing the likelihood of the model outputting the sequence $\mathcal{O}^* = (O_1^*, ...O_{N_s}^*)$. This corresponds to a product of probabilities of each state outputting the observed value. Logarithm of the computed likelihood is used to avoid very small numbers in models with a high number of states. Formally

$$L^* = \log\left(P(\mathcal{O}^*|\lambda)\right) = \sum_{i=1}^{N_s} \log\left(P(O_i^*|\lambda)\right), \tag{15}$$

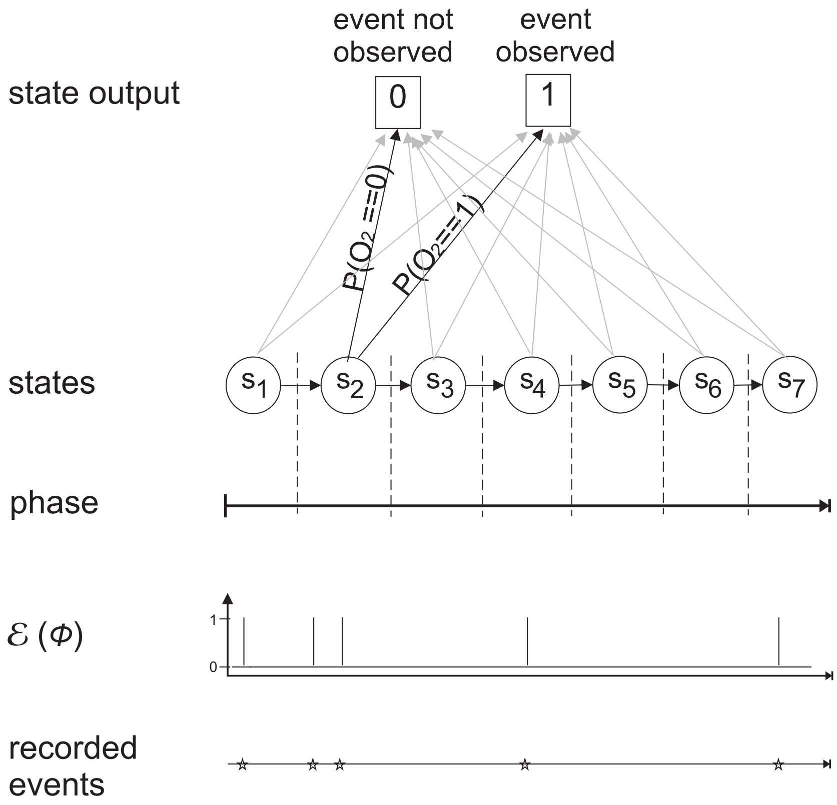

Figure 2: Overview of the proposed probabilistic model. It consists of discrete number of states, each corresponding to an interval of the action phase. As the phase progresses, so does the model in a left to right fashion through all the states. Upon visiting a state, a discrete symbol is output. The output symbol can be either 1, corresponding to semantic events being observed, or 0, corresponding to no events occurring during the particular interval, corresponding to the active state. The output probabilities are trained and evaluated according to indicator signal $\mathcal{E}$, which indicates observations of semantic events in the recorded data.

where

$$P(O_i^*|\lambda) = \begin{cases} B_{1,i}, & \text{if } O_i^* = 0 \\ B_{2,i}, & \text{if } O_i^* = 1 \end{cases}. \tag{16}$$

An observed action can then be classified as belonging to the action model for which L^* is the highest. Alternatively, if the likelihood is lower than some threshold value for all available models, we can conclude that the recorded action is something not observed before. This threshold can be selected by computing likelihoods for a number known sequences of a particular class (cross-validation set, not used in training) and taking the lowest obtained likelihood.

Note that for models in which the output probabilities of any state are 0 for observing an event, the resulting likelihood will be equal to zero for any examples with observed events in the particular state, regardless of potential matching in all other states. The same is true if the probability of not observing is zero, but the recorded sequence does contain such an observation. To avoid this problem, some threshold ϵ can be set for minimum and maximum values of $\mathbf{B}$, such that $B_{1,i} = \epsilon$, if $P(O_i = 0) < \epsilon$ and $B_{2,i} = 1 - \epsilon$, if $P(O_i = 1) > 1 - \epsilon$. If a cross-validation set is available, it should obviously be used to obtain this parameter as well. ϵ is chosen as the value, for which the performance on the cross-validation set is the best.

2.4. Verification with low level models

In order to distinguish action classes which exhibit similar patterns in terms of semantic events, lower level information can be used. We chose to utilize parametric hidden Markov models (PHMM) to encode execution trajectories for their discriminative, generative and generalization properties. PHMM is an extension to HMM, where the observation distributions depend on external parameters.

HMM is a Markov chain where each state corresponds to random output with some probability density. When representing motion trajectories the states are connected in a left-to-right structure and present discretized progression of time. State output corresponds to position coordinates, and is distributed according to a continuous Gaussian probability density function (PDF). The parameters (mean and covariance) of the density function are estimated during training with Baum-Welch algorithm [22]. In PHMMs, the means of the PDF are floating: they are dependent on external parameters via a linear mapping

$$\boldsymbol{\mu}_i(\boldsymbol{\theta}) = \mathbf{W}_i\boldsymbol{\theta} + \mathbf{b}_i, \tag{17}$$

where $\boldsymbol{\mu}_i$ is the mean of the PDF, $\mathbf{W}_i$ is a matrix of coefficients, $\mathbf{b}_i$ is y-intercept, i is state index and $\boldsymbol{\theta}$ are the external parameters. Both $\mathbf{W}_i$ and $\mathbf{b}_i$ are extracted from training data using a modified version of Baum-Welch algorithm for parametric hidden Markov models [30].

In our case, we defined the parameters $\boldsymbol{\theta}_j$ for trajectory j as average x, y and z coordinates of the trajectory. This way we captured spatial variations of the executions: the same model was used to apprehend executions of a given action irrespective of where on the table the action took place.

When using PHMMs for evaluation of unknown trajectories, the parameters $\boldsymbol{\theta}$ are first extracted from the data. The observation is then compared to all known models using the forward algorithm [22], where means of the state observations are calculated by Eq. 17 using the extracted $\boldsymbol{\theta}$. The matching between the unknown trajectory and a selected model is obtained by calculating the likelihood of the model outputting the exact same sequence of samples that forms the unknown trajectory. The trajectory can then be classified as belonging to the model with the highest likelihood.

PHMMs have been previously used for trajectory learning, recognition and reproduction in robotics [29, 32]. However, these authors only considered the recognition of presegmented sequences and did not use semantic relations between objects.

3. Evaluation

In this paper we evaluate the proposed approach on several manipulation tasks. These were milk pouring (20 examples), cereal pouring (23 examples), opening a box (21 examples), opening a bottle (20 examples), and finally, a peg-in-a-hole insertion task (12 examples). All recorded tasks consisted of three actions. First, the demonstrator reached for the object, then performed the main action, and finally withdrew from the object. See figures 3 and 7 for illustration of what the experiments looked like.

For semantic event detection we used RGB-D camera Kinect. Regions with uniform color and continuous 3-D surface were segmented from the background. This way we got a crude approximation of events between objects. Segment relations were then monitored and changes (touching, not-touching, appearing, disappearing) detected as explained in [10, Appendix 1]. This way we calculated function $E(t)$ as explained in Section 2, i. e. $E(t) = 1$ if a change happened at time t, and $E(t) = 0$ otherwise. See Figures 5 and 16 for typical examples of semantic event sequences in milk pouring and peg-in-a-hole tasks, respectively.

Execution trajectories were recorded with NDI 3dInvestigator motion capture system, which uses active markers to record 3-D trajectories. The data was modeled with PHMM as explained in Section 2.4. Rotation data was not used in the evaluation.

The evaluation consists of three parts. In Section 3.1 we present results for extraction of actions from longer observation sequences. In Section 3.2

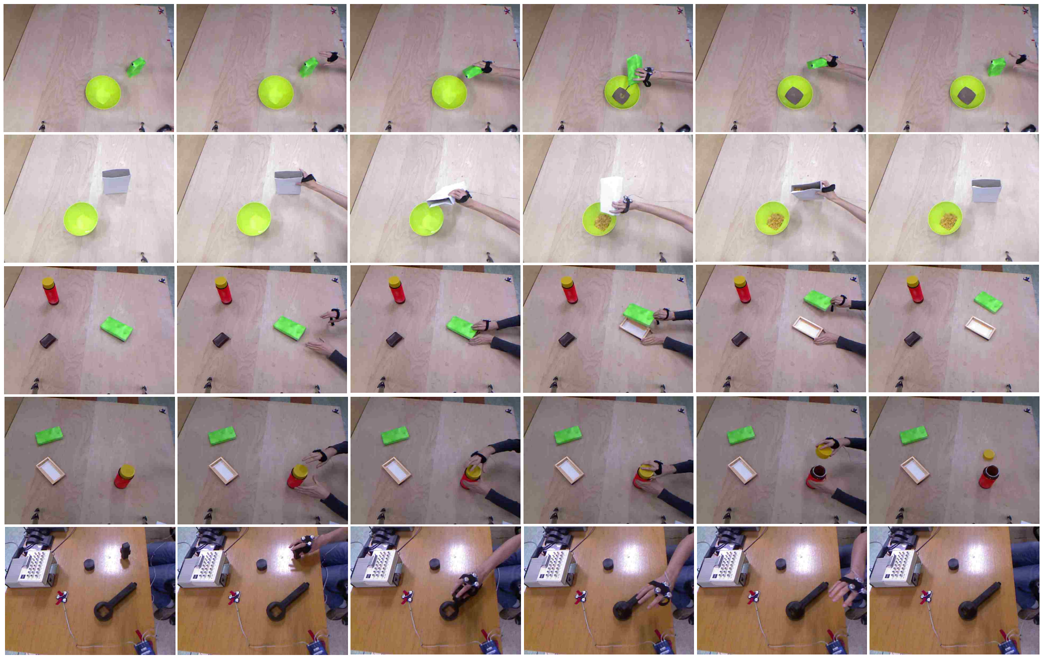

Figure 3: Video stills from the performed experiments. From top to bottom: pouring (chocolate) milk, pouring cereal, opening box, opening bottle, peg-in-a-hole.

we analyze general, all-vs-all action recognition properties of the novel probabilistic semantic models, similarity comparison between models using histogram distance measures, as well as recognition performance of lower level trajectory models. In Section 3.3 we compare phase mappings with uniform normalization of time and dynamic time warping.

3.1. Extraction of actions from observation

In this section we present results for top down search for actions in longer observations, such as the case where robot observes a human, looking for actions it already knows. This is a computationally expensive effort; for an observation sequence $\mathcal{O}$ of length, i.e. number of samples, T, there exist $\frac{T(T+1)}{2}$ possible subsequences. In order to locate an action, all these subsequences have to be checked with the action model, barring those exceeding some preset maximum or minimum subsequence length. In our case, all the subsequences were evaluated on the semantic level using the proposed probabilistic model first. Thirty best matches were passed to the lower level and evaluated using trajectory models. The subsequence with the best score was the final result.

The models were trained using roughly three quarters of randomly selected examples for each task. The segmentation of tasks into actions was done manually for the training set, as well as to provide ground truth for the test set. Number of states of the models was 10 for the semantic models and 50 for the PHMM.

3.1.1. Milk pouring

We first focus on the milk pouring example. Figure 7 shows stills from a typical recording of the milk pouring task and Figure 4 shows the trained semantic models. Clearly, the models are distinctive for each action. They can be easily interpreted. For the first action ("grasping the milk") there is low probability of observing events at the very beginning and higher probability in the second half of the action. The events in the states around the middle are due to manipulator's hand appearing in the scene; the higher values towards the end of the movement correspond to manipulator grasping the box. As the detection of touching from vision is unreliable in this case, the recorded events are spread over the last few states. Over the course of the second action ("pouring"), there is a lot of movement going on. The milk pack is taken to the bowl, rotated around, milk flowing, splashing, and so on. All this results in a stream of semantic events recorded throughout the whole action phase, as evident by non-zero probabilities for all states of the model. For the third action ("withdrawing"), we can see some events at the beginning, which correspond to the hand and the milk box ceasing to touch. The rest of the action is the hand moving away from the objects, which, typically, does not cause semantic events.

For evaluation, 5 examples were tested, which were not used during training. Testing all possible subsequences, probabilistic semantic models were used to identify those which had good similarity with the sought action. Due to the vast number ($\frac{T(T+1)}{2}$) of possible combinations, there are many subsequences that exhibit a given pattern of semantic events. Therefore, validation with low level features is needed in order the identify the best matching subsequence.

The thirty subsequences which were passed to the lower level can be seen in Figure 6. It can be seen that some of the selected subsequences are very similar. In fact, these subsequences have the same score according to the semantic models. This is due to the discrete nature of our model: moving the subsequence a couple samples left or right does not have any effect on

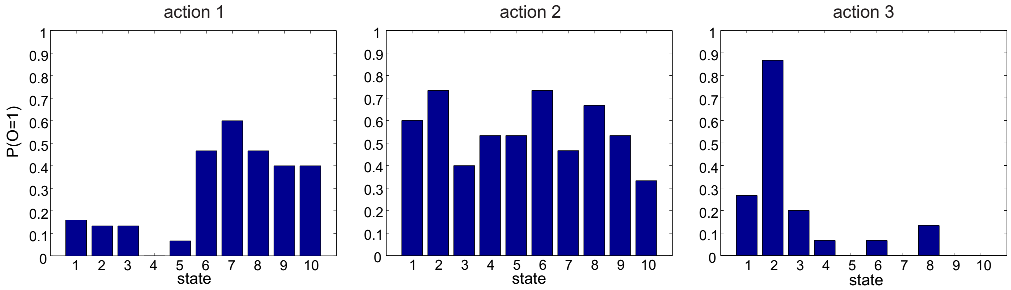

Figure 4: Trained models for the three actions comprising the milk pouring task. From left to right: grabbing the box, pouring, letting go of the box. The bars show probabilities of observing a semantic event in the given state.

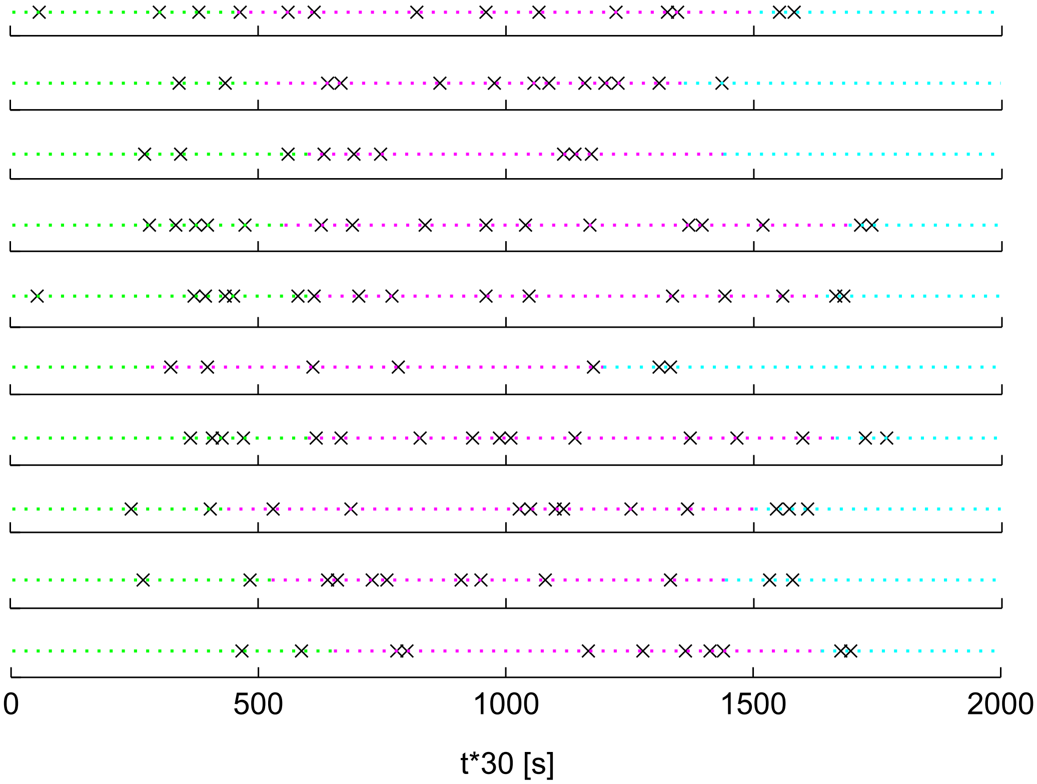

Figure 5: Semantic event sequences for 10 examples of the milk pouring task. The samples where $E(t) = 1$, signifying that a semantic event happened, are marked with 'x'. Green dots indicate grasping, magenta dots indicate pouring, and cyan dots indicate withdrawing actions.

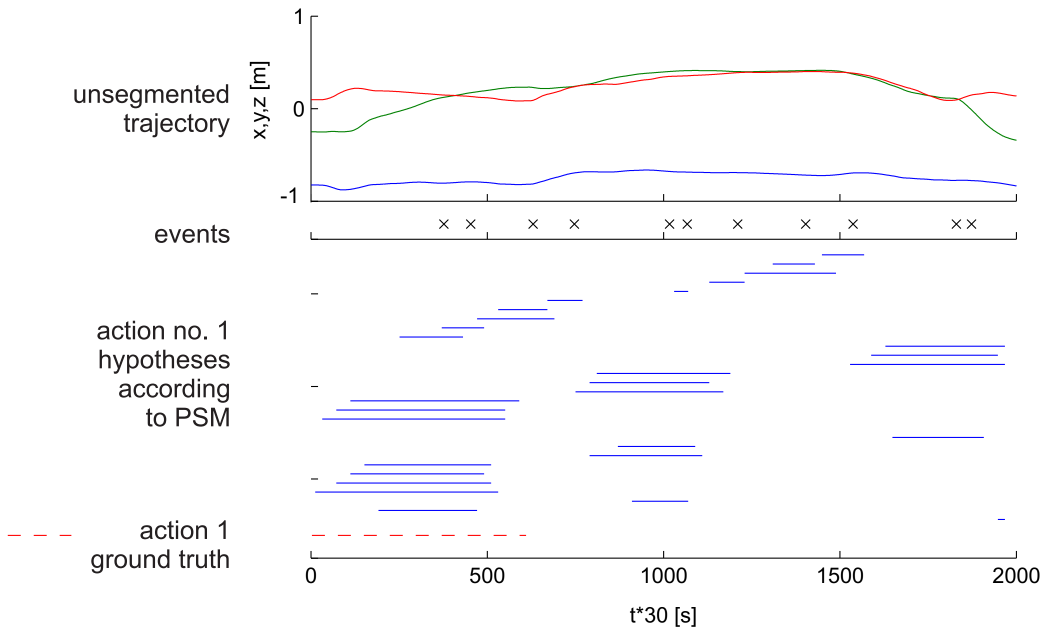

Figure 6: Action extraction hypotheses for grasping action in one of the milk pouring test examples. All possible subsequences of the recorded semantic event sequence during task observation (each event occurrence marked with black “x”) were evaluated with learned probabilistic semantic model (abbreviated PSM in the figure) for action number 1 (Fig. 4, left). Thirty best-matching subsequences are shown with blue lines. The graph shows that most of them identify parts of the recording, which exhibit the pattern of events being recorded towards the end. This is exactly what the model of the grasping action suggests.

its semantic score, as the relevant events fall into the same state. Therefore, the number of states used by the model defines the resolution of the action phase. The best subsequences were further evaluated with trajectory models (PHMM). The subsequence with the best score was then accepted as the final result.

The three basic actions of the task (grasping, pouring, withdrawing) were considered separately. This means that for each of them, evaluation with the action’s semantic model was followed by evaluation with the action’s PHMM. The obtained final best subsequence was then labeled as “the action”, whereas all other samples of the observation sequence were labeled as “not belonging to the action”.

The accuracy of the extracted segments for each evaluated case was calculated at the sample level as the ratio between samples that were classified

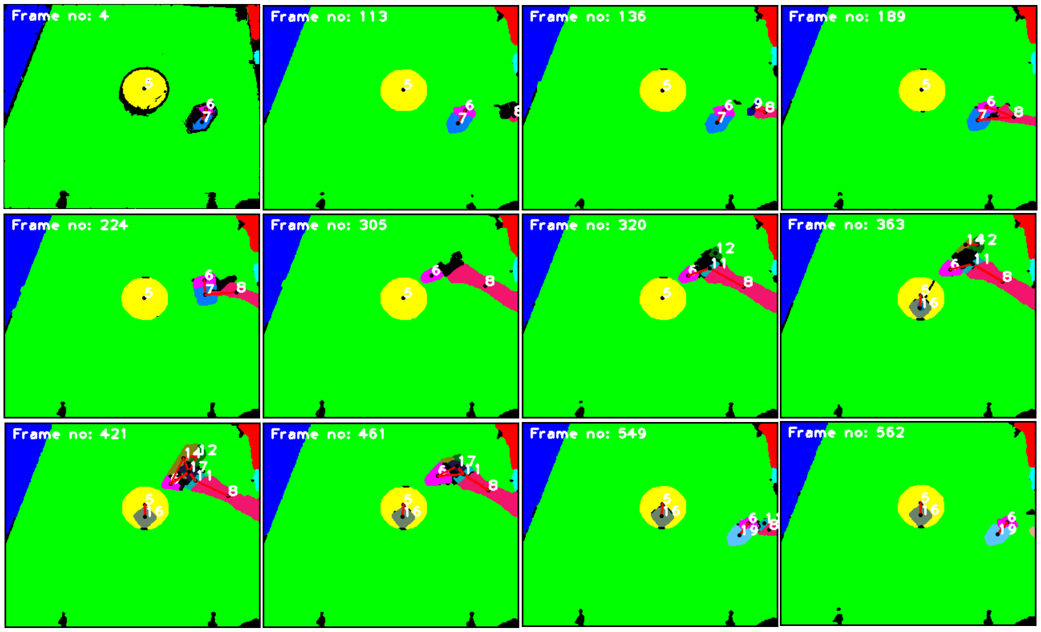

Figure 7: A sequence of images taken from observation of the milk pouring task, also shown in Figure 6. The frames were extracted at time instants when semantic events were recorded. Each frame corresponds to a consecutive event occurrence, marked with "x" in Figure 6. This graph shows that even in a simple milk pouring scene, which did not contain any clutter, the objects are not segmented perfectly. As a result, many semantic events are detected. For some of them it is not immediately clear which change in semantic relations triggered them (e. g. frames no. 189 and 224 show the same semantic situation). In fact, only five "real" semantic events happened during task execution: hand appeared, hand touched the milk box, milk appeared in the bowl, hand released the box, hand disappeared. These correspond to frames 113, 189, 363, 549 and 562, respectively. All other semantic events are consequences of noise.

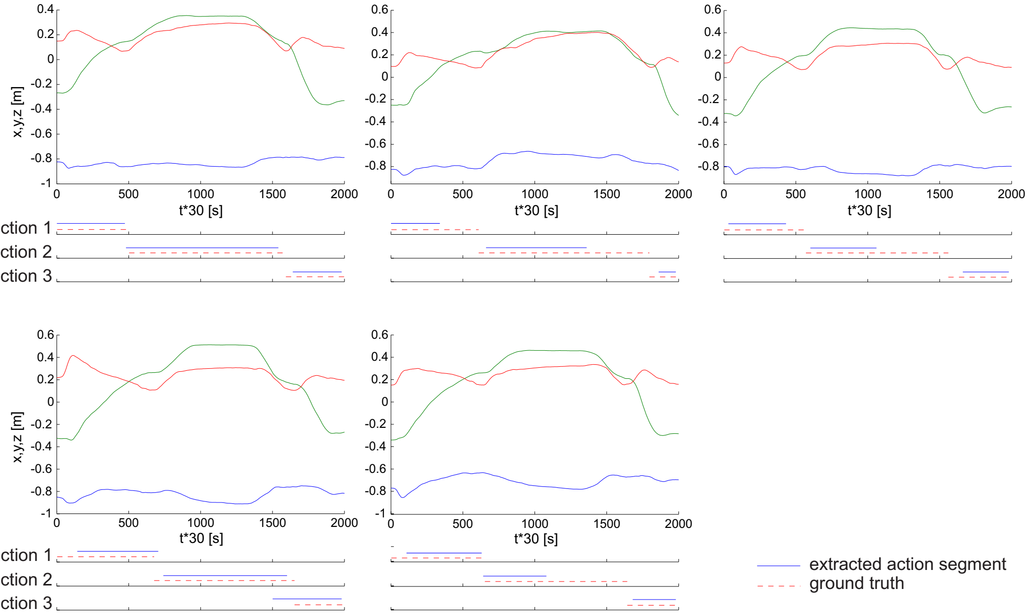

Figure 8: Results for the milk pouring task. The graphs show Cartesian trajectories of the execution; the blue lines at the bottom of each graph show calculated segments, corresponding to individual actions.

correctly and the total number of recorded samples in the observed sequence. The final result was computed by averaging across all of the tested examples:

$$\text{Accuracy} = \frac{1}{N_{ex}} \sum_{i=1}^{N_{ex}} \frac{1}{N_a^i} \sum_{j=1}^{N_a^i} \frac{N_{j,i}^{correct}}{T_i}, \tag{18}$$

where N_{ex} stands for the number of tested examples (recordings) and N_a^i denotes the number of actions in the example i. $N_{j,i}^{correct}$ denotes the number of samples in recording i that were correctly classified when evaluating action j. Note that true positives as well as true negatives are included in $N_{j,i}^{correct}$. T_i denotes the number of samples in recording i. Ground truth sample labels were provided manually. Figure 8 shows the results. It is evident that the extracted segments overlap with the ground truth. The accuracy for the milk pouring task was 0.90.

3.1.2. Cereal pouring

Next, we switched milk with cereal. This allows for an interesting comparison, as cereal is much less homogeneous in appearance and thus the

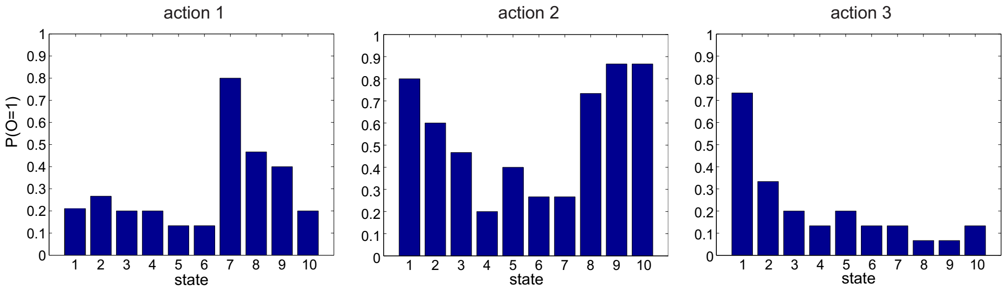

Figure 9: Trained models for the three actions of the cereal pouring task. From left to right: grasping the cereal box, pouring, letting go of the box. The bars show probabilities of observing a semantic event in the given state.

results of image segmenting algorithm are even more unpredictable. As a consequence, the learned semantic models for the pouring action are different compared to the milk pouring, as evident by comparing Figures 4 and 9. For the "grasping" and "withdrawing" actions, on the other hand, the models are similar.

In total, 23 examples of the cereal pouring task were recorded and 8 of them used as a test set. Results are shown in Figure 10. The average accuracy (as per Eq. (18)) for cereal pouring task was 0.92.

3.1.3. Opening a box

In this experiment, demonstration started with a chocolate box, sitting on the table. The demonstrator then opened the box using both of his hands and put the lid on the table next to the box. Finally, the demonstrator withdrew his hands from the scene. The learned semantic models can be seen in Figure 11. Comparing with the "grasping" and "withdrawing" models of the pouring tasks, we can notice that they are similar. Higher probabilities in the second part of the grasping action are likely the result of using both hands to reach for the box as opposed to the pouring tasks, where just one hand was used. This caused a higher number of events.

Extraction results for the box opening task are shown in Figure 12. The average accuracy was 0.89 for the test set, which was composed of 6 out of 21 recorded examples.

3.1.4. Opening a bottle

Semantically, box opening and bottle opening are similar tasks. However, the way in which the task is accomplished, is different. Opening a box

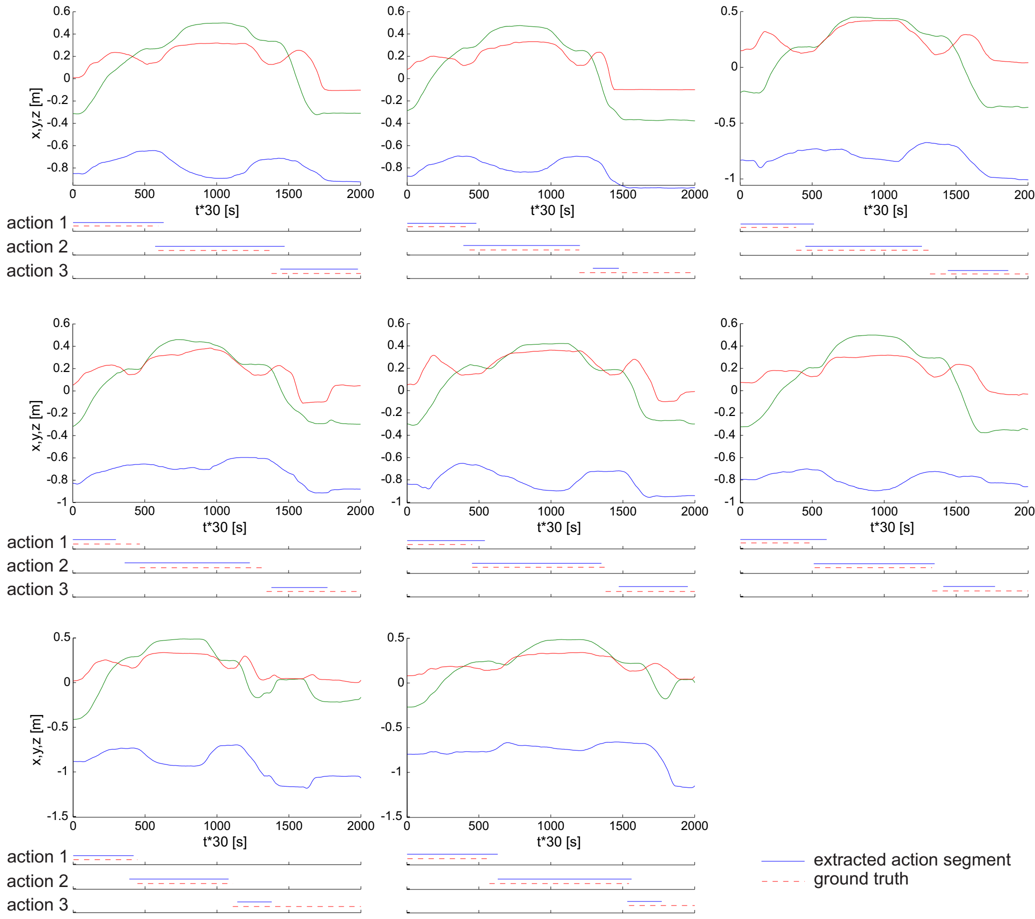

Figure 10: Results for the cereal pouring task. The graphs show Cartesian trajectories of the execution; the blue lines at the bottom of each graph show calculated segments, corresponding to individual actions. As can be seen, the system had some trouble extracting action number 3; in almost half of the examples the calculated segment is significantly shorter then the ground truth, shown with a dotted line. Nevertheless, the result is approximately correct.

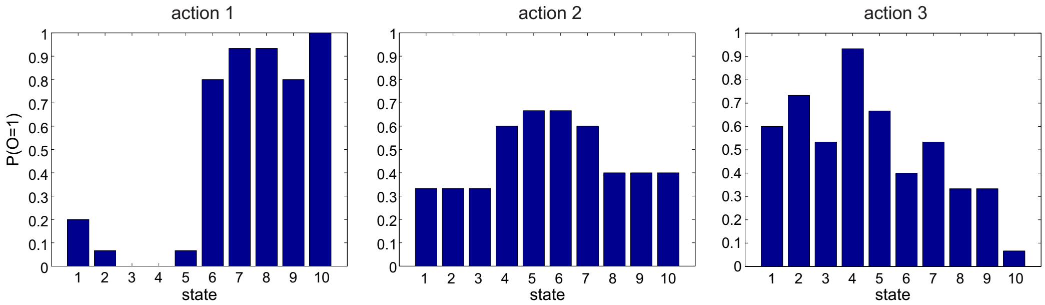

Figure 11: Trained models for the three actions of the box opening task. From left to right: grasping the box, opening, letting go of the box and the lid. The bars show probabilities of observing a semantic event in the given state.

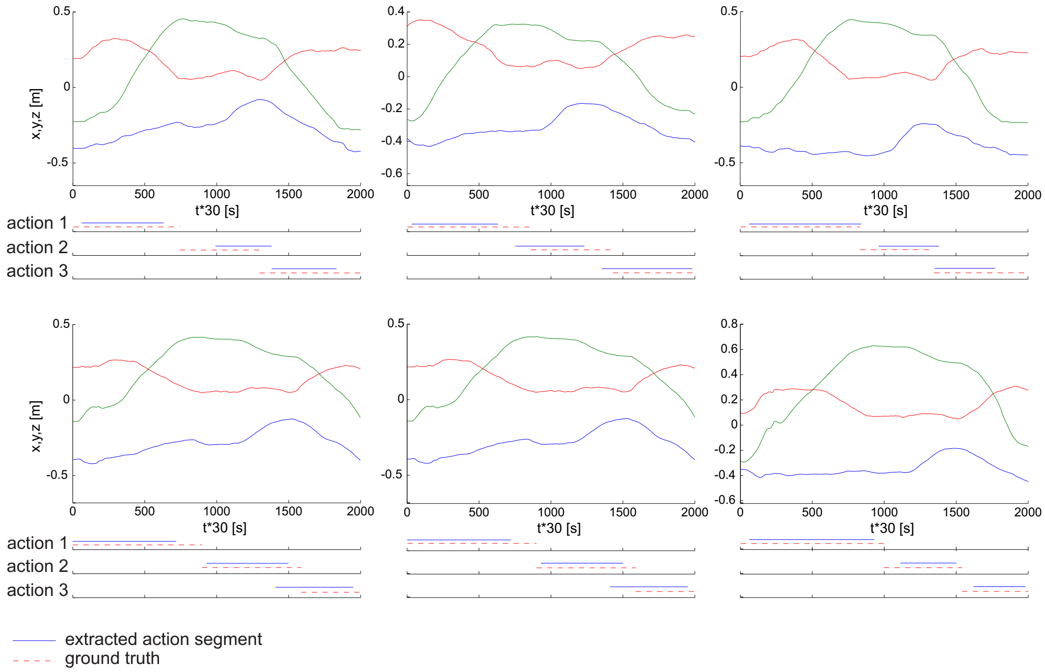

Figure 12: Results for the box opening task. The graphs show Cartesian trajectories of the execution; the blue lines at the bottom of each graph show calculated segments, corresponding to individual actions. The performance is comparable to other tasks.

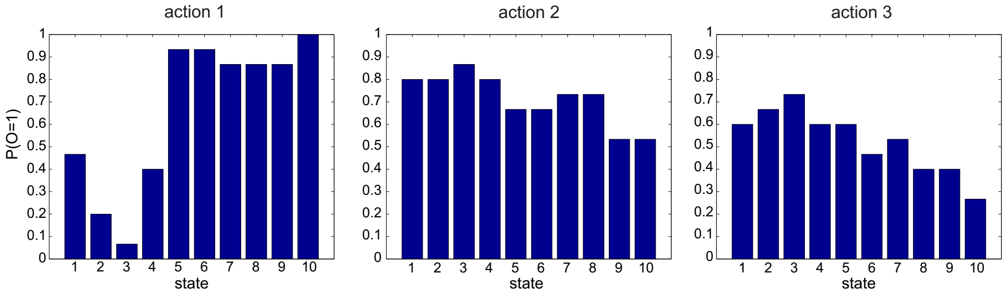

Figure 13: Trained models for the three actions of the bottle opening task. From left to right: grasping the bottle, unscrewing the cap, letting go of the bottle and the cap. The bars show probabilities of observing a semantic event in the given state.

consists of picking up the lid and placing it aside, while the bottle's cap needs to be unscrewed. Naturally, this means different execution trajectories. However, comparing middle graphs in Figures 11 and 13, it can be seen that probabilistic semantic models for the respective actions (opening and unscrewing) are also somewhat different. Namely, the unscrewing action model shows high probabilities of semantic event detection all across the phase, which is the result of unscrewing motion of the hand. The hand is in constant movement, touching and un-touching the bottle at multiple points. Such movement poses a big problem for the image segmentation algorithm, which results in noisy detection of semantic events.

For this task, 20 examples were recorded, out of which 15 were used for training and 5 for the test set. Action extraction results with the learned models are shown in Figure 14, with the average accuracy of 0.89.

3.1.5. Peg in a hole

Finally, we performed an experiment with demonstrator showing execution of the "peg-in-a-hole" task. In this task, the demonstrator was asked to perform a step of Cranfield assembly benchmark [33], which consists of a square peg being put into a quadratic hole. In total, 11 recordings of this experiments were made, 8 of which were used for training and 3 for testing. As opposed to other experiments, there was much more clutter on the table, such as cables and instruments. There were also more variations in execution, as the demonstrator sometimes used a different hand to help insert the peg. Figure 16 shows ten semantic event sequences for the task. Compared to Figure 5, which shows events in milk pouring examples, the peg-in-a-hole examples exhibit much more noise. As can be seen in Figure 15, which shows

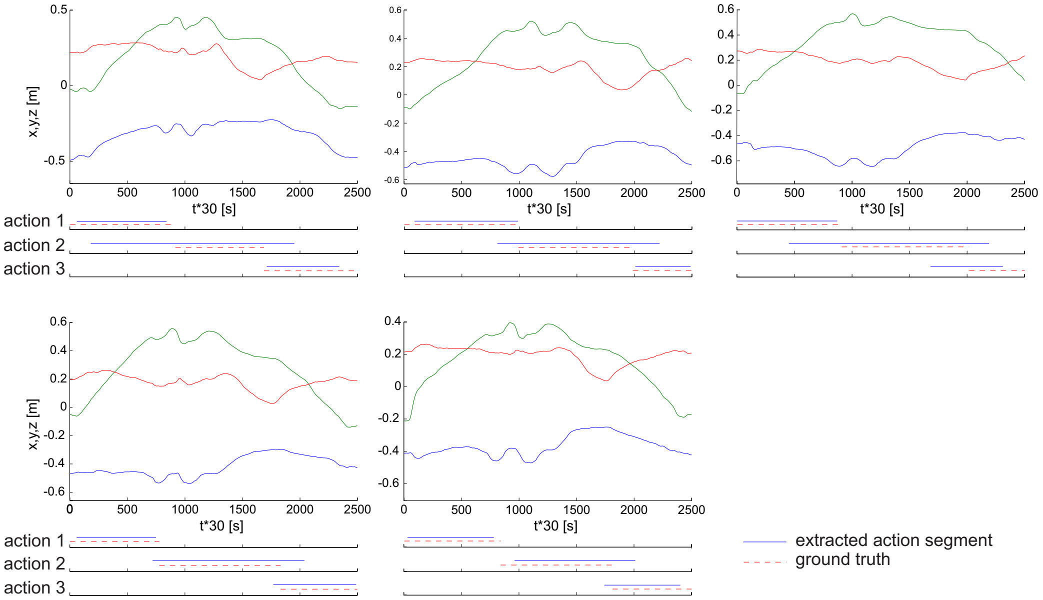

Figure 14: Results for the box opening task. The graphs show Cartesian trajectories of the execution; the blue lines at the bottom of each graph show calculated segments, corresponding to individual actions. The system had some trouble extracting segment number 2, overshooting it in most of the examples.

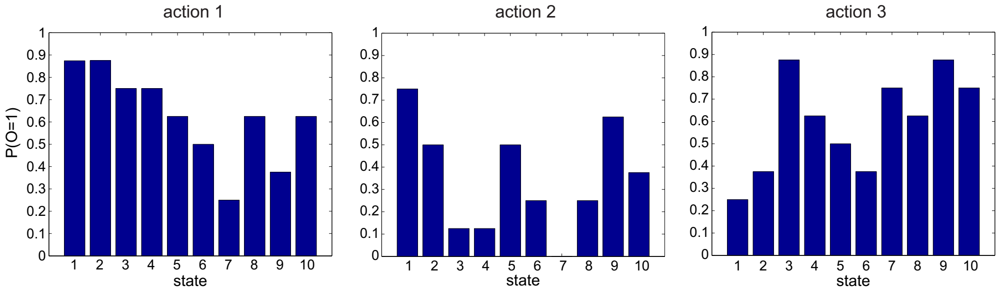

Figure 15: Trained models for the three actions of the peg in a hole task. From left to right: grasping the peg, inserting the peg into the hole, letting go of the peg. The bars show probabilities of observing a semantic event in the corresponding states.

the trained models, the increased measurement noise results in high output probabilities all across the execution. This is especially evident in the model for the third ("withdrawing") action, which demonstrates a shape with two peaks, which is unique compared to similar actions from other tasks. Still, the models are clearly action-discriminative.

Figure 17 shows action extraction results for peg in a hole experiment. The extracted segments correspond relatively well to the ground truth despite noisy measurements. The resulting overall average accuracy for the peg in a hole task is 0.89.

3.2. Cross task comparison of actions with probabilistic semantic models

To further evaluate the performance of the proposed probabilistic semantic models we provide comparison of actions with models from different tasks. The actions were segmented manually and the resulting sequences were compared to all of the models and the likelihood computed according to eq. (15). No trajectory data was used in this test.

The resulting confusion matrix is shown in Figure 18. Each row corresponds to a recording (sequence of semantic events) of an action. Each column corresponds to a probabilistic semantic model of the given action. The models and the recordings are grouped by tasks, e. g. the first three models and the first 60 recordings come from the milk pouring tasks.

The color of each cell represents the likelihood of event sequence in the corresponding row being output by the model of the corresponding column. Warmer colors represent higher likelihoods, while colder colors represent lower likelihoods. The likelihood was obtained by averaging the result of all possible models built with the training set consisting of approximately

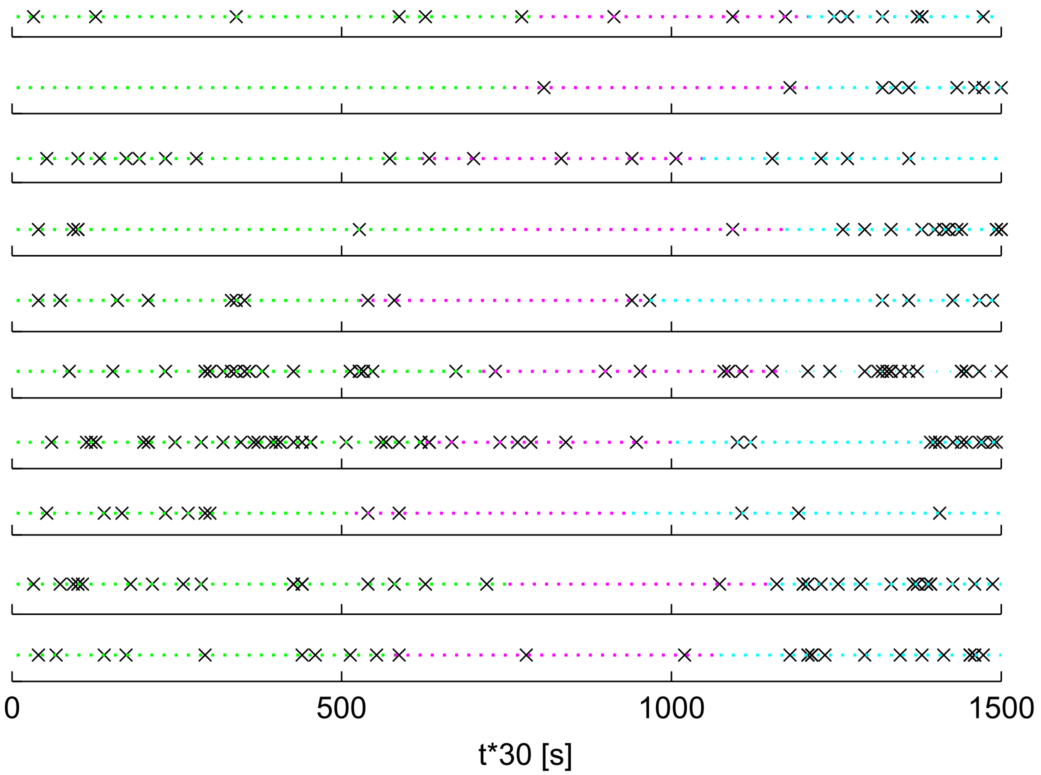

Figure 16: Semantic event sequences for 10 examples of the peg-in-a-hole task. The samples where $E(t) = 1$, signifying that a semantic event happened, are marked with 'x'. Green dots indicate grasping, magenta dots indicate inserting the peg, and cyan dots indicate withdrawing actions.

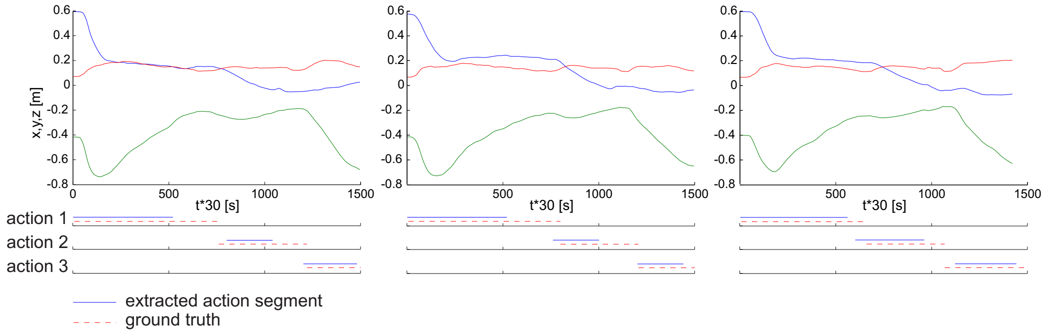

Figure 17: Results for peg in a hole task. The graphs show Cartesian trajectories of the execution; the blue lines at the bottom of each graph show calculated segments, corresponding to individual actions. The noise in measurements results in slightly lower performance compared to previous experiments.

three quarters of the remaining examples for the particular action. For example, there are 21 box-opening examples. For each one example, there exist $\binom{20}{15} = 15504$ possible training sets consisting of 15 out of the remaining 20 examples. All the possible 15504 models are trained and used to evaluate the likelihood of the current example. The likelihood shown in Figure 18 is the average of all the obtained likelihoods for the example.

The emerging pattern shows the properties of the proposed method. Most importantly, the diagonal elements exhibit warmer colors than the rest of the matrix, meaning that comparing an action with its own model we get high likelihood. Another interesting result is high likelihood of actions with respect to models of similar actions, but from different tasks. For example, grasping action from box opening task gets a high likelihood also for the grasping models from the milk pouring and bottle opening tasks. This behavior is, however, not completely consistent. For example, the likelihood of the grasping model of the peg-in-a-hole task is quite low for the same box-grasping examples.

Some of the examples also show high likelihood where the model and the action clearly do not belong together. For example, "placing into hole" model gives high likelihood for cereal pouring and withdrawing actions.

In summary, our results show that probabilistic semantic models provide basic recognition of actions. However, as there are false positives, they need to be used in combination with lower level models based on trajectory data. In combination they provide a powerful action extraction and recognition mechanism. This confirms what was already noted in Sections 2.3 and 2.4.

3.2.1. Histogram distance based comparison of models

Apart from using the learned semantic models to evaluate unknown observations, the models themselves can be compared to one another. This opens the possibility for the robot to reason about semantic differences between actions and combine models which are determined to be similar.

Methods for histogram comparison can be used for this purpose. Different metrics have been previously proposed, such as Quadratic-Form [34], Chi-Squared [35], or Earth Mover's distance [36]. Figure 19 shows the confusion matrix obtained when comparing the models by the Earth Mover's method. Interestingly, the result suggests that similar models according to this metric are the grasping actions of milk pouring compared to grasping of both opening tasks, withdrawing actions of the opening tasks, as well as box opening, bottle opening, milk pouring, and peg grasping actions. In a

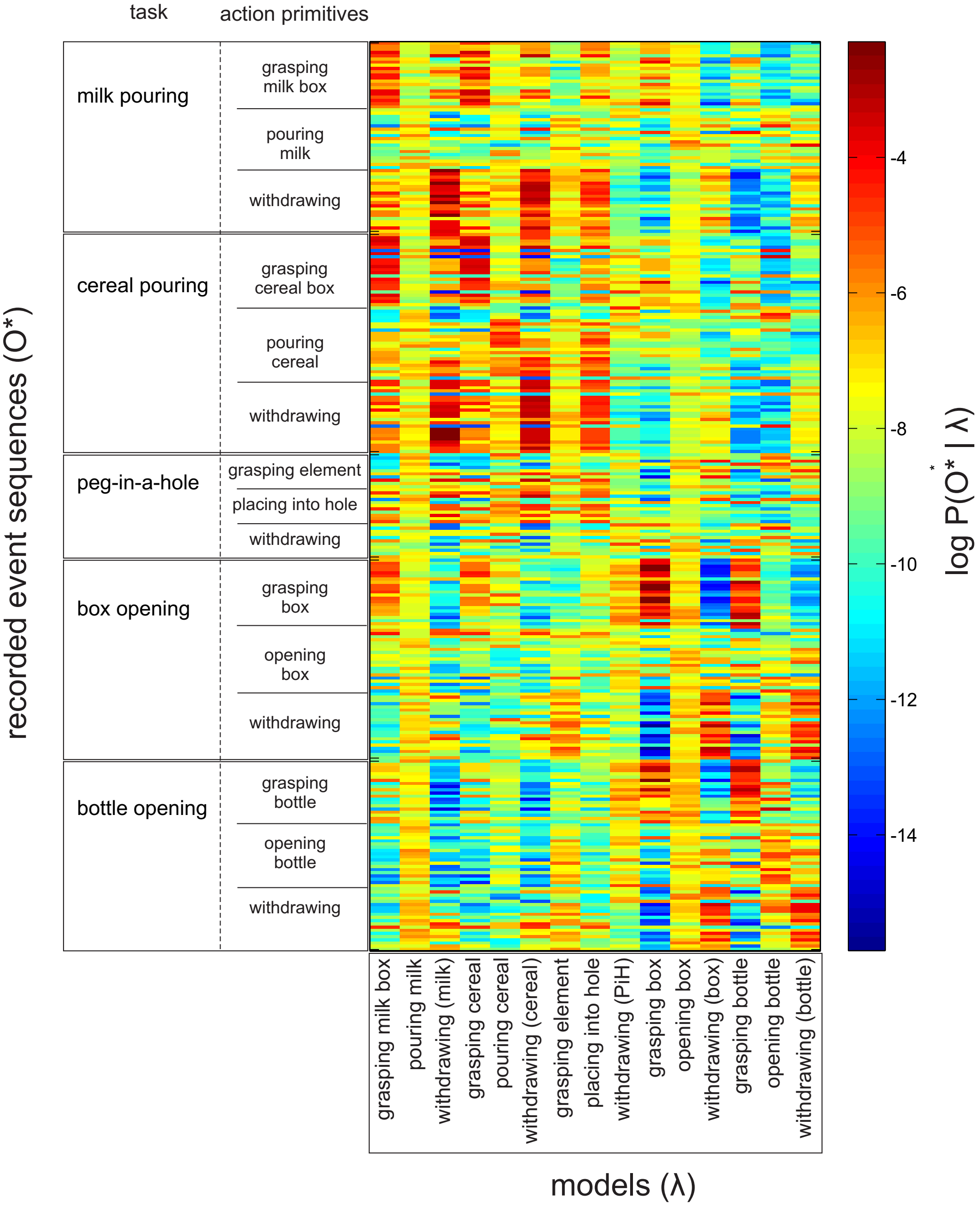

Figure 18: Recognition of actions across all recorded tasks. The recordings were presegmented and probabilistic semantic models trained for each class. Color of each cell in the matrix corresponds to the likelihood of a given model outputting the given sequence, calculated according to Eq. (15).

way, this corresponds to results obtained in Section 3.2, where, for example, many “grasping box” recordings in Figure 18 received high likelihoods for the model of “grasping milk”. On the other hand, this does not seem to be the case for the “grasping bottle” recordings, which received generally lower likelihoods for the same case, even though the corresponding case in Figure 19 suggests that their models should be similar.

The biggest problem with histogram comparison is that there is no definite metric which would be the best to utilize. It must be chosen by the user according to the problem at hand. One must thus be careful when qualitatively evaluating the differences between histograms, as comparisons with different metrics produce different results.

3.2.2. Cross task comparison of actions with trajectory models

Here we present results for comparison of the actions based on trajectory data only. This is a hard problem, as even humans have trouble differentiating between actions of another person based on their trajectories without other context. Consider for example, a game of pantomime, which is non-trivial in spite of exaggerations in the demonstrator's movement. In manipulation tasks, the differences between trajectories are much more subtle. Figure 20 illustrates performance of the low-level PHMM models in our experiments. Due to computational complexity of the training and testing algorithms, it was not possible to perform the exhausting evaluation with all training set combinations in the same way as with the semantic models. Instead, the training and testing set were chosen randomly with $\frac{3}{4}$ of examples belonging to training set and $\frac{1}{4}$ to testing set. As you can see, the trajectories generally received high likelihoods for the corresponding “correct” models, however there are many cases where high likelihoods were calculated for completely unrelated recording-model pairs. This confirms that comparison of actions based on trajectories alone would not be sufficient in this case, as the context of objects plays a key role.

3.3. Dynamic Time Warping

In Section 2 we presented a way to normalize demonstration sequences to a common phase. The mapping defined in Eq. 2 uniformly shrinks the sequences into the specified phase interval. This approach does not take into account the fact that demonstrations may vary not only in their duration, but also in their temporal profile. For example, the same action can be executed quickly in the beginning and slowly towards the end, or vice versa. Dynamic

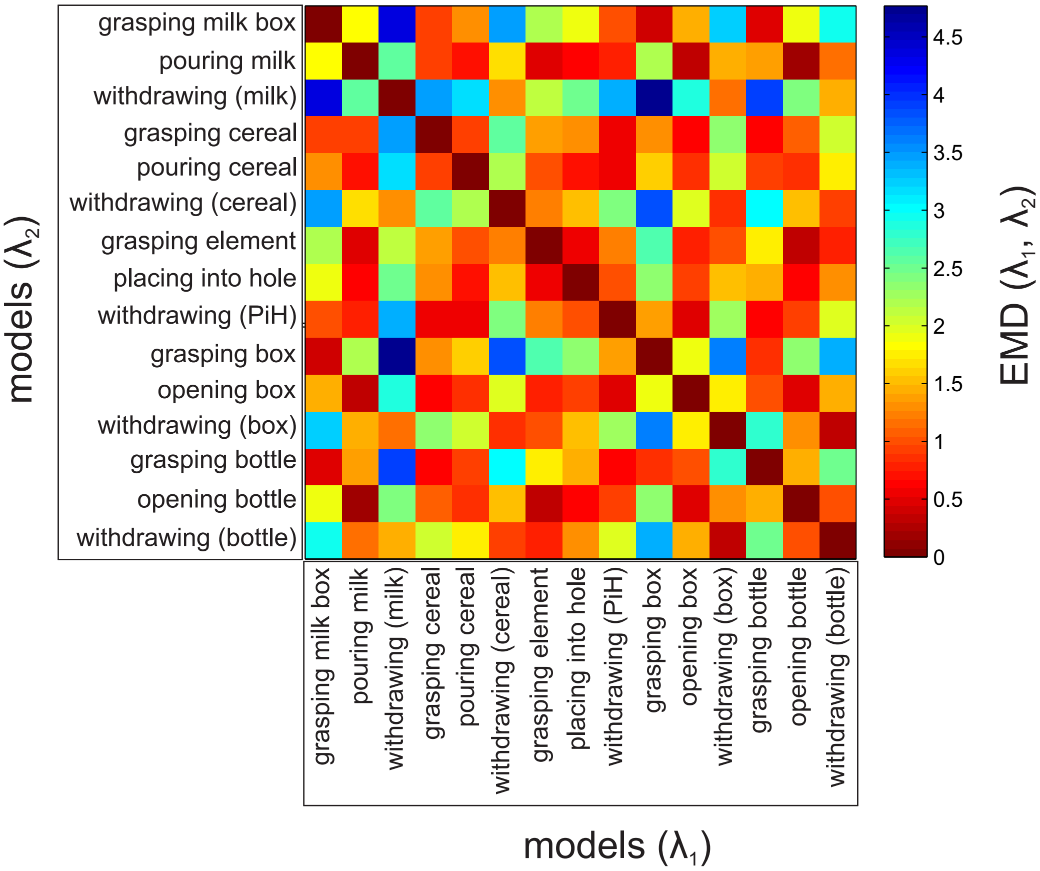

Figure 19: Earth Mover's distance between models of all actions. Warmer colors represent lower distance and higher similarity, while colder colors correspond to higher distance and thus lower similarity between a pair of models.

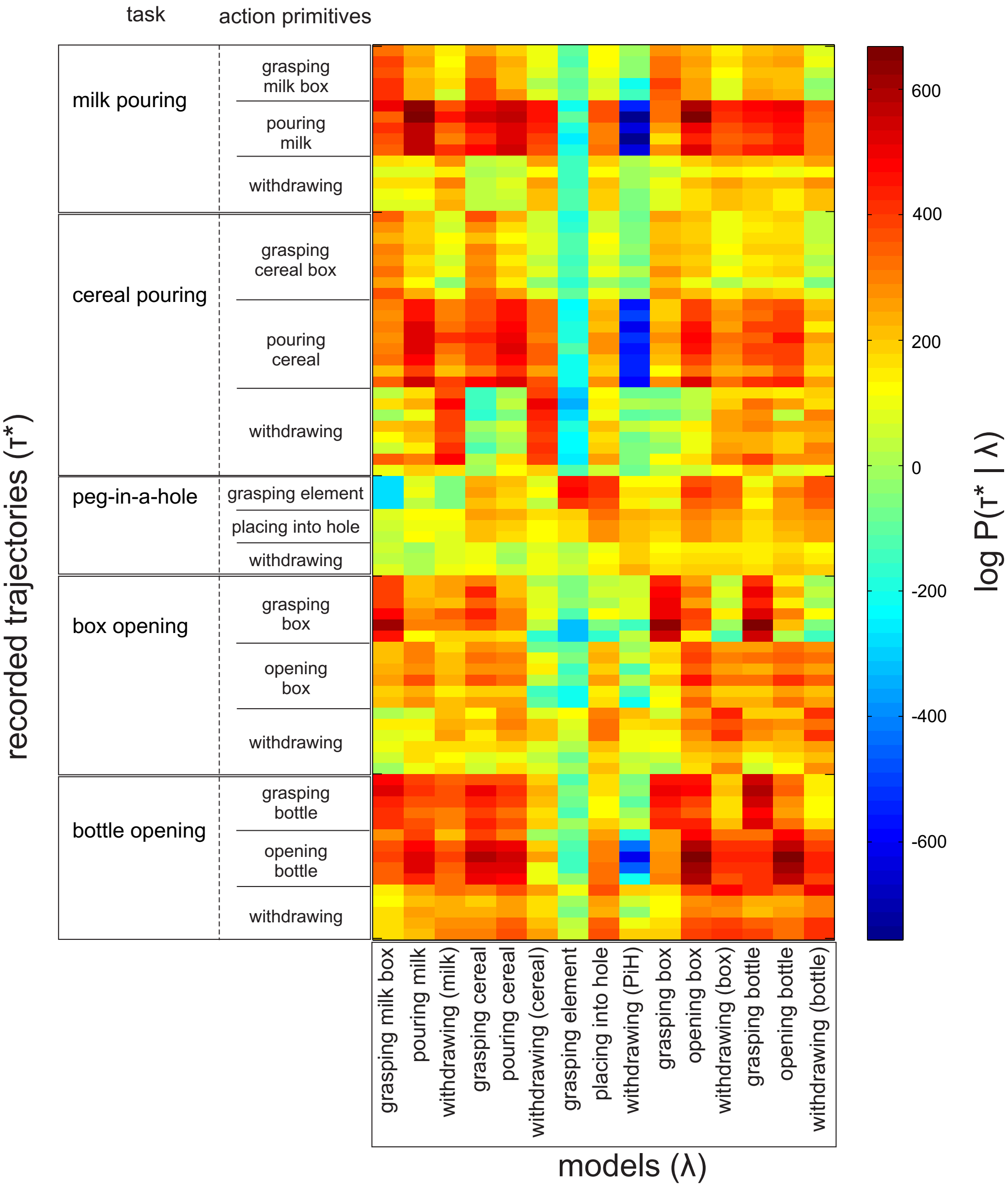

Figure 20: Recognition of execution trajectories using parametric hidden Markov models. Higher likelihood (warmer colors) corresponds to stronger matching, whereas colder colors correspond to weaker matching.

time warping (DTW, [31]) is a technique which can be used to align two sequences by warping their temporal profile. It has been previously applied to motion trajectories [37][38]. However, the standard DTW approach suffers from quadratic computational complexity and can only be used for alignment of two sequences. For these reasons, Zhou et. al [39] presented Generalized time warping (GTW), which does not have these drawbacks. GTW defines the temporal profile of a trajectory as a combination of non-linear functions and finds the optimal solution by optimizing their weights.

In our case, GTW can be applied to demonstration trajectories in order to obtain a warped temporal profile for each example. This profile is then copied to semantic event sequence function E and transformed into the required $[0, 1]$ interval. Before testing, unknown actions are aligned using the same principle. This way, it also becomes important "where" the events occurred rather than just "when", as events occurring around similar parts of paths get grouped together. See Figure 21 for illustration, which shows original and time-warped case for 15 examples of the grasping action of the milk pouring task. The upper graph shows execution trajectories, the middle one sequences of semantic events, while the resulting models are shown in the bottom graph. Clearly, you can see that the trajectories in the right graph are lined up compared to the original case, as there is less empty space between individual lines. This resulted in slightly different placement of some of the semantic events on the action phase. For example, the events of examples number 13 and 15, which were in the third discrete bin in the original case moved to the fourth bin in the warped case. Such differences resulted in slightly different semantic models being trained on the respective sets. Earth mover's distance between the two histograms is 0.39, which is rather small.

Figure 22 shows results of recognition for both of the semantic models. The figure shows log likelihood obtained when testing all available recordings (except those used for training) with the obtained models - the left graph shows results for the original case, while the right graph shows results for the aligned case. As you can see, there is not much difference in the performance of both models.

These results indicate that, for the experiments presented in this paper, phase mapping with uniform scaling of time (Eq. (2)) is sufficient. The difference in temporal alignment of detected events is largely compensated for by discretization of phase, and thus the difference between the original model and the GTW-aligned model is too small to have an impact on recognition performance, let alone justify the extra computational expense.

4. Discussion and conclusion

In this paper we presented a hierarchical approach capable of extracting known actions from observation sequences. The main novelty of our approach is the higher level representation with probabilistic semantic models. These are calculated from temporal sequences of semantic events, which denote changes in semantic relations between objects.

Semantic events are extracted from machine vision. This process is intrinsically unreliable and prone to noise. The main feature of the proposed models is that the events are modeled probabilistically, and thus robustness against noise is achieved. Detection of semantic events using a simple system based on uniform color and shape properties of objects proved to be effective enough to result in semantic models with good discriminative properties.

The models are built by calculating the probability that execution of a given action results in detection of semantic events during a particular interval of the action phase. We do not take into account information about which objects triggered the event, nor what type of a relational change happened. The only important information is *when* during the action phase the event occurred.

This is in contrast to the semantic event chains, as presented in [10]. They annotate semantic relations for each pair of objects separately, where objects are approximated with image segments and each image segment is identified with a label. The temporal information is preserved partially: it is evident from a chain in what sequence the events happened.

The work by Luo et al. [18] also deals with semantic event chains and noise. They transform the chains into semantic string representations and then use string kernels to evaluate similarity between strings of different recordings. Interestingly, they also make the event chains less descriptive in the process. The semantic strings are constructed by collapsing rows in the event chains to only contain the changes without repeating relations. This way, information about which event happened before or after another is lost for all events that are not caused by the same pair of image segments. It could thus be argued that by achieving robustness to noise, we neglect the association of semantic events to specific object or image segment pairs, while they neglect the temporal aspect.

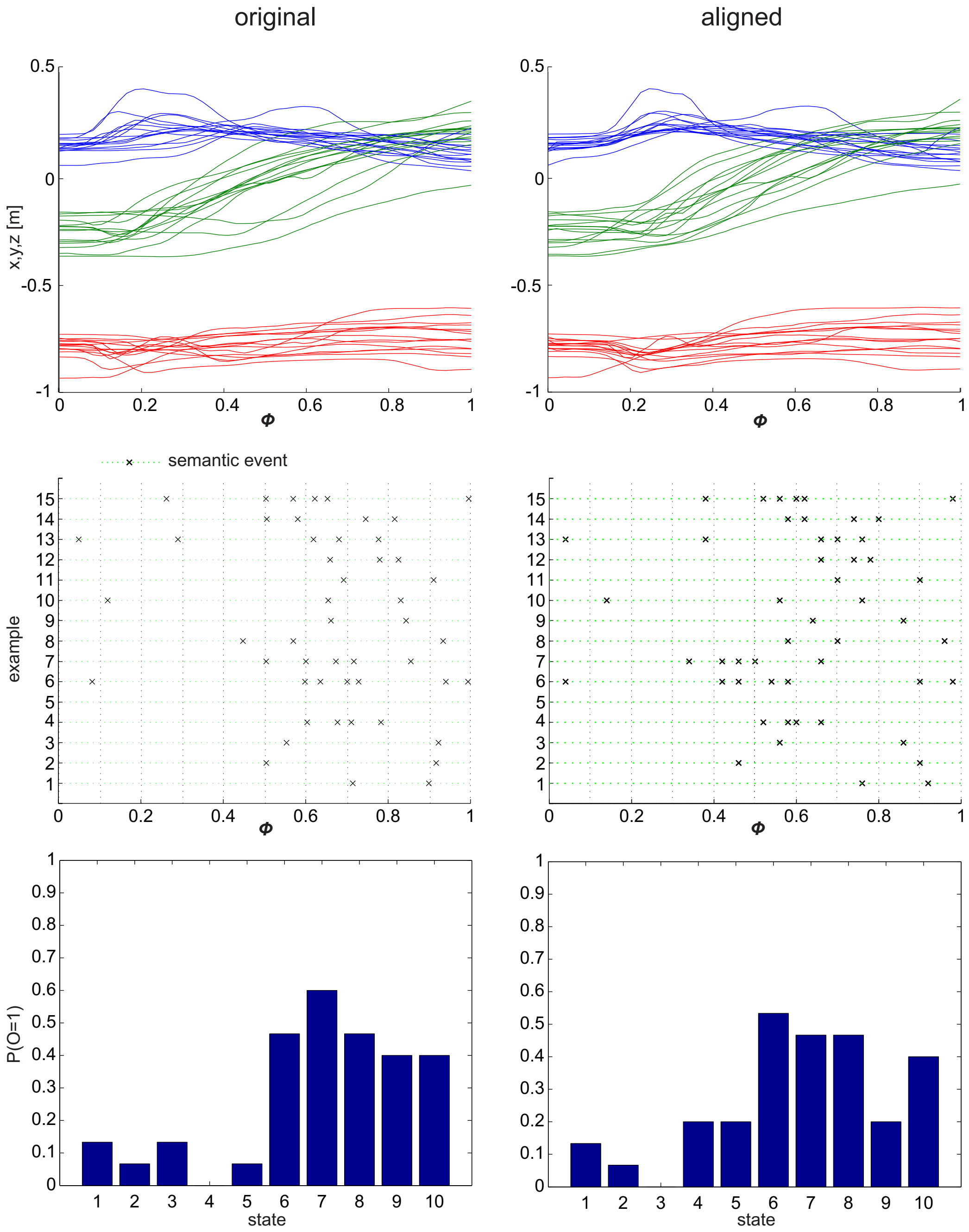

Figure 21: Time warping for grasping action of the milk pouring task. The left side shows the original case, while the right side shows the result after GTW was applied for temporal alignment. The top graphs show trajectories of 15 examples, the middle graphs show semantic events of the same examples, and the bottom graphs show the trained semantic models. The alignment resulted in slight difference between the models.

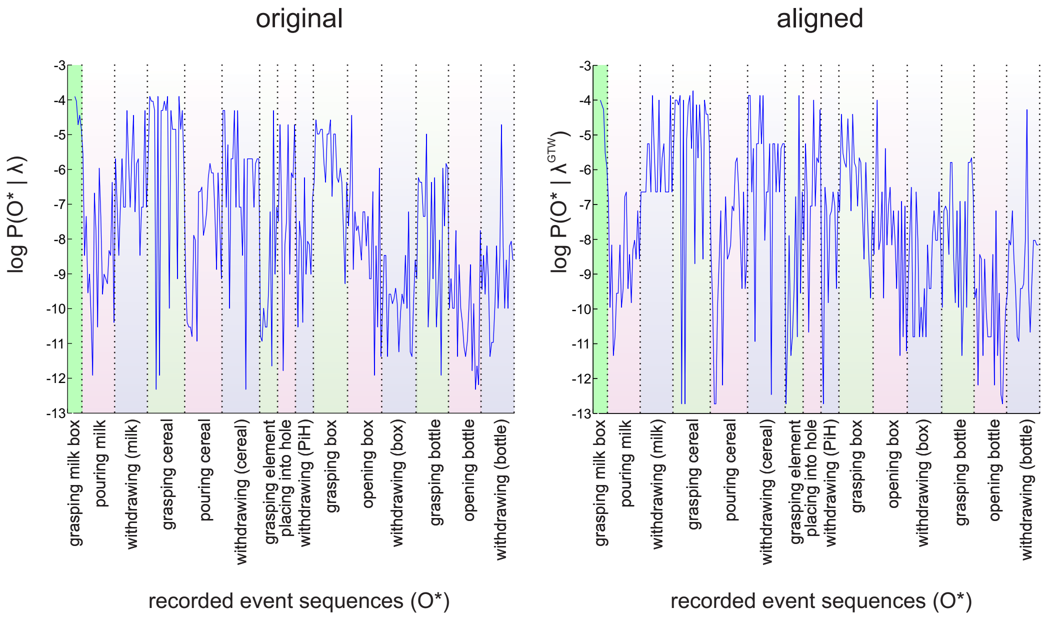

Figure 22: Recognition performance for grasping action of the milk pouring task. The left graph shows the original case, while the right graph shows the result for the model obtained after GTW was applied. The models were trained using 15 examples of grasping action from the milk pouring task, shown in Figure 21. The leftmost green shaded area on both graphs consists of the testing set of the same milk-grasping action, while other green shaded areas correspond to grasping actions from other tasks. Similarly, red and blue shaded areas consist of second and third actions of each task, respectively. As you can see, the likelihoods obtained for the original and the aligned model exhibit a rather similar pattern.

The probabilistic semantic models are therefore less descriptive compared to event chains or semantic strings in terms of association of semantic events to specific object or image segment pairs; they are however more descriptive from the temporal standpoint, as they carry information about likelihood of occurrences of events for all parts of the action phase.

The fact that the proposed probabilistic semantic models are defined in the action phase domain allows for easy coupling with arbitrary low level models. In this paper we utilized parametric hidden Markov models to encode motion trajectories as lower level action representation. This allowed us to perform action extraction from observation sequences in two stages. First, the upper level model was used to search for subsequences which exhibited the desired pattern in terms of semantic events. The subsequences with the best fit were then used as hypotheses for the trajectory level and evaluated to find the final result. The reasoning behind this approach is that the semantic models are much faster at evaluating unknown sequences than Markov models. Computational complexity of evaluating an unknown trajectory (subsequence) with PHMM is $O(M_s^2 N)$, where N is number of samples in the subsequence and M_s number of states in the model. The average length N of all subsequences of a sequence of length T is $\frac{T+2}{3}$, therefore the complexity of the whole search process with PHMM is $O(M_s^2 T^3)$. Typically, the number of states M_s would be in the tens, while T could be in the hundreds or thousands. On the other hand, the proposed semantic models only evaluate the score of a sequence at the discrete states, corresponding to phase intervals. Therefore, summation over the states is all that is needed. The complexity of the search process with the semantic models is thus $O(T^2 N_s)$, where N_s is the number of states of the semantic model.

On the hand, the trajectory models are needed because many actions can be semantically similar, yet different. These are differentiated by their execution trajectories. Note, however, that full-scale recognition using trajectories only is also difficult in case of manipulation actions. Nevertheless, they proved to be discriminative enough to distinguish between hypotheses selected from the sequences with good match at the semantic level.

Another important aspect in favor of using both a semantic as well as trajectory model in a hierarchical way is that in manipulation tasks, the changes in semantic relations usually relate to the intent of the manipulator's actions. This way, by combining probabilistic models of semantic events with lower level features, such as execution trajectories, we bring together the effects (i. e. changes in semantic relations) and their cause (the movement) in a common generative framework.

The number of states in the proposed action model is a hyperparameter, which needs to be set for each action. Note that while the results of our experiments were good even without putting much effort into the choice of the hyperparameter, the selection could be relatively simply automated by cross validation techniques [40]. However, a bad choice for the number of states could lead to a diminished performance. For this reason, one of our future focuses will be on the development of continuous formulation of the action representation, using for example kernel density estimation techniques [41, 42] to approximate event distributions. This way, division of phase into states would no longer be needed and event observation likelihood would be represented with a probability density function defined over the whole phase. Such models would certainly have some advantages. For example, in the present, discrete case, it is not important whether one or several events are detected in a single phase interval. Events that occur in sequences quicker than the resolution of the phase space permits are lost. This would not be the case with the continuous approach. Note, however, that kernel density estimation methods generally require some hyperparameters to be set as well.

In the experiments presented in this paper uniform normalization of time was used to transform recordings into phase space. Time warping techniques can be used to compensate for temporal variations within sequences. However, in our experiments we did not find much inequality between models trained from uniformly normalized data and time warped data.

Future prospective work includes expanding the approach to semantically higher levels on one hand and towards the robot execution on the other hand. Probabilistic semantic models are higher in abstraction than execution specific features, but still short of a full scale representation of task semantics, such as semantic event chains. We believe that probabilistic semantic models can be used to help with the construction of semantic event chains, because the major problem in their construction from real time observation data is oversegmentation due to vision noise. Probabilistic semantic models could be used to reason about recorded events' likelihood and therefore reject erroneous events.

Probabilistic semantic models could also be utilized to support the robot's own action execution. As they provide a model of the intended consequences of a manipulation action, they can be used as a metric of success. By monitoring semantic events in the scene during execution, the parts of the action phase where the robot's movement results in different semantic events than expected can be quickly identified. This way, the robot can autonomously identify which portion of the execution trajectory needs to be adapted (e. g. by using reinforcement learning [43]) in order to improve the result of the manipulation.

References

[1] F. Lv, R. Nevatia, Recognition and segmentation of 3-D human action using HMM and multi-class AdaBoost, in: Proceedings of the 9th European Conference on Computer Vision - Volume Part IV, ECCV'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 359–372.

[2] M. Hoai, Z.-Z. Lan, F. De la Torre, Joint segmentation and classification of human actions in video, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, 2011, pp. 3265–3272.

[3] M. C. Silaghi, Spotting subsequences matching an HMM using the average observation probability criteria with application to keyword spotting, in: Proceedings of the 20th National Conference on Artificial Intelligence, Pittsburgh, Pennsylvania, 2005, pp. 1118–1123.

[4] S. B. Kang, K. Ikeuchi, Toward automatic robot instruction from perception-temporal segmentation of tasks from human hand motion, IEEE Transactions on Robotics and Automation 11 (5) (1995) 670–681.

[5] O. Brock, J. Kuffner, J. Xiao, Motion for manipulation tasks, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer, Berlin, Heidelberg, 2008, pp. 615–645.

[6] T. Lozano-Pérez, Spatial planning: A configuration space approach, IEEE Transactions on Computers C-32 (1983) 108–120.

[7] K. Ikeuchi, T. Suehiro, Toward an assembly plan from observation, part I: Task recognition with polyhedral objects, IEEE Trans. Robotics and Automation 10 (3) (1994) 368–385.

[8] J. Xiao, Automatic determination of topological contacts in the presence of sensing uncertainties, in: IEEE International Conference on Robotics and Automation (ICRA), Atlanta, Georgia, 1993, pp. 65–70.

[9] X. Ji, J. Xiao, Automatic generation of high-level contact state space, Int. J. Robotics Research 20 (1999) 238–244.

[10] E. E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, F. Wörgötter, Learning the semantics of object-action relations by observation, Int. J. Rob. Res. 30 (10) (2011) 1229–1249.

[11] F. Wörgötter, E. E. Aksoy, N. Kruger, J. Piater, A. Ude, M. Tamosiunaite, A simple ontology of manipulation actions based on hand-object relations, IEEE Transactions on Autonomous Mental Development 5 (2) (2013) 117–134.

[12] Y. Yang, C. Fermüller, Y. Aloimonos, Detection of manipulation action consequences (MAC), in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, Oregon, 2013, pp. 2563–2570.

[13] H. Kjellström, J. Romero, D. Kragić, Visual object-action recognition: Inferring object affordances from human demonstration, Computer Vision and Image Understanding 115 (1) (2011) 81–90.

[14] B. Yao, L. Fei-Fei, Modeling mutual context of object and human pose in human-object interaction activities, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, 2010, pp. 17–24.

[15] H. Kjellström, J. Romero, D. Martinez, D. Kragić, Simultaneous visual recognition of manipulation actions and manipulated objects, in: D. Forsyth, P. Torr, A. Zisserman (Eds.), Computer Vision - ECCV 2008, Vol. 5303 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2008, pp. 336–349.

[16] M. Sridhar, A. G. Cohn, D. C. Hogg, Learning functional object-categories from a relational spatio-temporal representation, in: Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence, IOS Press, Amsterdam, The Netherlands, The Netherlands, 2008, pp. 606–610.

[17] E. E. Aksoy, M. Tamosiunaite, R. Vuga, A. Ude, C. Geib, M. Steedman, F. Wörgötter, Structural bootstrapping at the sensorimotor level for the fast acquisition of action knowledge for cognitive robots, in: IEEE

Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL), 2013, pp. 1–8.

[18] G. Luo, N. Bergström, C. H. Ek, D. Kragic, Representing actions with kernels, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2011, San Francisco, CA, USA, September 25-30, 2011, 2011, pp. 2028–2035.

[19] B. Dellen, E. E. Aksoy, F. Wörgötter, Segment tracking via a spatiotemporal linking process including feedback stabilization in an n-d lattice model, Sensors 9 (11) (2009) 9355–9379.

[20] S. Perera, N. Barnes, A simple and practical solution to the rigid body motion segmentation problem using a RGB-D camera, in: Proceedings of the 2011 International Conference on Digital Image Computing: Techniques and Applications, Washington, DC, USA, 2011, pp. 494–500.

[21] D. Schiebener, J. Morimoto, T. Asfour, A. Ude, Integrating visual perception and manipulation for autonomous learning of object representations, Adaptive Behavior 21 (5) (2013) 328–345.

[22] L. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.

[23] D. Paul, Speech recognition using hidden markov models, Lincoln Laboratory Journal 3 (1) (1990) 41–62.

[24] J. Hu, S. G. Lim, M. K. Brown, Writer independent on-line handwriting recognition using an {HMM} approach, Pattern Recognition 33 (1) (2000) 133 – 147.

[25] S. Eickeler, A. Kosmala, G. Rigoll, Hidden markov model based continuous online gesture recognition, in: Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on, Vol. 2, 1998, pp. 1206–1208.

[26] F. Bashir, A. Khokhar, D. Schonfeld, Object trajectory-based activity classification and recognition using hidden markov models, Image Processing, IEEE Transactions on 16 (7) (2007) 1912–1919.

[27] C. Sminchisescu, A. Kanaujia, Z. Li, D. Metaxas, Conditional models for contextual human motion recognition, in: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, Vol. 2, 2005, pp. 1808–1815.

[28] A. Lpez-Mndez, J. R. Casas, Model-based recognition of human actions by trajectory matching in phase spaces, Image and Vision Computing 30 (11) (2012) 808 – 816.

[29] D. Herzog, A. Ude, V. Krüger, Motion imitation and recognition using parametric hidden Markov models, in: 2008 8th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Daejeon, Korea, 2008, pp. 339–346.

[30] A. D. Wilson, A. F. Bobick, Parametric hidden Markov models for gesture recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (9) (1999) 884–900.

[31] L. Rabiner, B. H. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993.

[32] V. Krüger, D. Herzog, S. Baby, A. Ude, D. Kragic, Learning actions from observations, IEEE Robotics Automation Magazine 17 (2) (2010) 30–43.

[33] K. Collins, A. J. Palmer, K. Rathmill, The development of a European benchmark for the comparison of assembly robot programming systems, in: K. Rathmill, P. MacConail, S. Oleary, B. J. (Eds.), Robot Technology and Applications, Springer-Verlag, New York, 1985, pp. 187–199.

[34] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, W. Niblack, Efficient color histogram indexing for quadratic form distance functions, IEEE Trans. Pattern Anal. Mach. Intell. 17 (7) (1995) 729–736.

[35] O. Pele, M. Werman, The quadratic-chi histogram distance family, in: Computer Vision ECCV 2010, Vol. 6312 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 749–762.

[36] Y. Rubner, C. Tomasi, L. J. Guibas, The earth mover's distance as a metric for image retrieval, Int. J. Comput. Vision 40 (2) (2000) 99–121.

[37] F. Zhou, F. De la Torre Frade, Canonical time warping for alignment of human behavior, in: Advances in Neural Information Processing Systems Conference (NIPS), 2009, pp. 2286–2294.

[38] D. Gong, G. Medioni, Dynamic manifold warping for view invariant action recognition, in: Computer Vision (ICCV), 2011 IEEE International Conference on, 2011, pp. 571–578.

[39] F. Zhou, F. De la Torre, Generalized time warping for multi-modal alignment of human motion, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR '12, IEEE Computer Society, Washington, DC, USA, 2012, pp. 1282–1289.

[40] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 1137–1143.

[41] B. W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman & Hall, London, 1986.

[42] R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger, J. Piater, Learning grasp affordance densities, Paladyn Journal of Behavioral Robotics 2 (1) (2011) 1–17.

[43] R. S. Sutton, A. G. Barto, Introduction to Reinforcement Learning, 1st Edition, MIT Press, Cambridge, MA, USA, 1998.