Model-Free Incremental Learning of the Semantics of Manipulation Actions

Eren Erdal Aksoy[a,*], Minija Tamosiunaite[a], Florentin Wörgötter[a]

[a] *Georg-August-Universität Göttingen, BCCN, Department for Computational Neuroscience Inst. Physics-3, Friedrich-Hund Platz 1, D-37077 Göttingen, Germany*

Abstract

Understanding and learning the semantics of complex manipulation actions are intriguing and non-trivial issues for the development of autonomous robots. In this paper, we present a novel method for an on-line, incremental learning of the semantics of manipulation actions by observation. Recently, we had introduced the Semantic Event Chains (SECs) as a new generic representation for manipulations, which can be directly computed from a stream of images and is based on the changes in the relationships between objects involved in a manipulation. We here show that the SEC concept can be used to bootstrap the learning of the semantics of manipulation actions without using any prior knowledge about actions or objects. We create a new manipulation action benchmark with 8 different manipulation tasks including in total 120 samples to learn an archetypal SEC model for each manipulation action. We then evaluate the learned SEC models with 20 long and complex chained manipulation sequences including in total 103 manipulation samples. Thereby we put the event chains to a decisive test asking how powerful is action classification when using this framework? We find that we reach up to 100% and 87% average precision and recall values in the validation phase and 99% and 96% in the testing phase. This support the notion that SECs are a useful tool for classifying manipulation actions in a fully automatic way.

*Corresponding author
Email address: eaksoye@physik3.gwdg.de (Eren Erdal Aksoy)

Preprint submitted to Robotics and Autonomous Systems *July 9, 2014*

1. Introduction

One of the main problems in cognitive robotics is how to recognize and learn human demonstrations of new concepts, for example learning a relatively complex manipulation sequence like cutting a cucumber. Association-based or reinforcement learning methods are usually too slow to achieve this in an efficient way. They are therefore most often used in combination with supervised learning. Especially the Learning from Demonstration (LfD) paradigm seems promising for cognitive learning ([1, 2, 3, 4, 5]) because humans employ it very successfully. The problem that remains in all these approaches is how to represent complex actions or chains of actions in a generic and generalizable way allowing inferring the essential "meaning" (semantics) of an action irrespective of its individual instantiation.

In our earlier studies we introduced the "Semantic Event Chain" (SEC) as a possible descriptor for manipulation actions [6, 7]. The SEC framework analyzes the sequence of changes of the *spatial relations* between the objects that are being manipulated by a human or a robot. Consequently, SECs are invariant to the particular objects used, the precise object poses observed, the actual trajectories followed, or the resulting interaction forces between objects. All these aspects are allowed to change and still the same SEC is observed and captures the "essence of the action" as demonstrated in several action classification tests performed by us [6, 7, 8, 9].

In this paper, we address the problem of on-line, incremental learning of the semantics of manipulation actions observed from human demonstrations. We use the concept of SECs as the main processing tool to encode manipulations in a generic and compact way. Manipulations are continuous in the temporal domain but with event chains we discretize them by sampling only decisive key time points. Those time points represent topological changes between objects and the hand in the scene which are highly descriptive for a given manipulation. Our main intent here is to design a cognitive agent that can infer and learn frequently observed spatiotemporal changes embedded in SECs in an unsupervised manner whenever a new manipulation instance occurs. The learning phase is bootstrapped only with the semantic similarities between SECs, i.e. the encoded spatiotemporal patterns, without using any prior knowledge about actions or objects. Since we use computer vision methods to derive event chains, our approach for incremental learning of semantics is highly grounded in the signal domain. To the best of our knowledge, this is the first attempt to apply reasoning at the semantic level, while being fully grounded at the signal level, to learn manipulations with an unsupervised method. Note, here – on purpose – we do not include any object- or other information to show the power of our methods to fully automatically and in an unsupervised way extract action and object information. Clearly, in praxis, it will often make sense to include whatever additional knowledge is available to further ease action understanding.

The paper is organized as follows. We start with introducing the state of the art. We next provide a detailed description of each processing step; extraction of SEC representations and learning model-SECs for each observed manipulation. In the next section, we discuss experimental results from the proposed framework, which also includes validation and testing of the learned models. We finally conclude with a discussion.

2. State of the Art

Learning from Demonstration (LfD) has been successfully applied both at the control [1, 2, 10] as well as the symbolic level [3, 4, 5]. Although various types of actions can be encoded at the control level, e.g. trajectory-level, this is not general enough to imitate complicated actions under different circumstances. On the other hand, at the symbolic level, sequences of predefined abstract action units are used to learn complex actions, but this might lead to problems for execution as many parameters are left out in a symbolic representation. Although our approach with SECs is a symbolic-level representation, SECs can be enriched with additional decisive descriptors (e.g. trajectory, pose, etc.) and do not use any assumption or prior knowledge in the object or action domain. Ideas to utilize relations to reach semantics of actions can be found as early as in 1975. For instance, [11] introduced the first approach about directed scene graphs in which each node identifies one object. Edges hold spatial information (e.g., LEFT-OF, IN-FRONT-OF, etc.) between objects. Based on object movement (trajectory) information, events are defined to represent actions. The main drawback of this approach is that the continuous perception of actions is ignored and is substituted instead by idealized hand-made image sequences. This approach, however, had not been pursued in the field any longer as only now powerful enough image processing methods became available from which object and action information can be extracted.

Still there are only a few approaches attempting to reach the semantics of manipulation actions in conjunction with the manipulated objects [12, 13, 14, 15, 16]. The work in [12] is one of the first approaches in robotics that uses the configuration transition between objects to generate a high-level description of an assembly task from observation. Configuration transitions occur when a face-contact relation between manipulated and stationary environmental objects changes. The work presented in [13] represents an entire manipulation sequence by an activity graph which holds spatiotemporal object interactions. The difficulty is, however, that very complex and large activity graphs need to be decomposed for further processing. In the work of [14], segmented hand poses and velocities are used to classify manipulations. A histogram of gradients approach with a support vector machine classifier is separately used to categorize manipulated objects. Factorial conditional random fields are then used to compute the correlation between objects and manipulations. Visual semantic graphs (inspired from our scene graphs) were introduced in [15] to recognize action consequences based on changes in the topological structure of the manipulated object. In [16] activity trees were presented to recognize actions using a minimal action grammar. Recent works such as [17] modeled human activities employing the human skeleton information as well as roles of manipulated objects. Although all those works to a certain extent improve the classification of manipulations and/or objects, none of them extracts key events of individual manipulations or learns a descriptive semantic model to represent different manipulation tasks.

In this sense, to our best knowledge, our work is the first study to evaluate and learn the semantics of manipulations in an incremental and model free manner. The concept of semantic event chains has been successfully utilized and extended by others [18, 19, 20, 21, 22, 23] not only to represent manipulation actions but also to replicate them by robots. The work in [18] presented active learning of goal directed manipulation sequences, each was recognized using semantic similarities between event chains. Our scene graphs were represented with kernels in [19] to further apply different machine learning approaches. Additional trajectory information was used in [20] to reduce noisy events occur in SECs. Others [21, 22, 23] showed execution of various manipulations with different robots by using the key spatiotemporal points provided by SECs.

3. Method

In this method section we will present the core algorithmic components where are complex details will only be given in the Appendix. This should make reading easier, while still everything is present to implement this algorithm if desired.

3.1. Data Acquisition

In this work, we investigate eight different manipulation actions: *Pushing*, *Hiding*, *Putting*, *Stirring*, *Cutting*, *Chopping*, *Taking*, and *Uncovering*. Fig. 1 (a) shows a sample frame for each manipulation action. All movies used in this study can also be found at `www.dpi.physik.uni-goettingen.de/~eaksoye/MANIAC_DATASET`. The *Pushing* action shows how a hand can move objects around randomly. In the action of *Hiding*, some objects are made invisible by covering them with other objects. In the *Putting* action objects are taken from the supporting background and put on top of each other. The *Stirring* action represents a scenario in which a spoon is used to stir some liquid in a bucket. In the *Cutting* action, a hand is cutting vegetables by moving a cutting tool back and forth. In the *Chopping* action, a cutting tool follows a straight trajectory to divide vegetables into parts. The *Taking* action represents a scenario where some objects are taken down and put on the supporting background. In the *Uncovering* action some objects are becoming visible after moving occluding objects away.

We recorded 15 different versions for each of these manipulations by asking 5 different individuals to demonstrate each manipulation 3 times with different objects in various scene contexts. Fig. 1 (b) depicts a sample frame from each

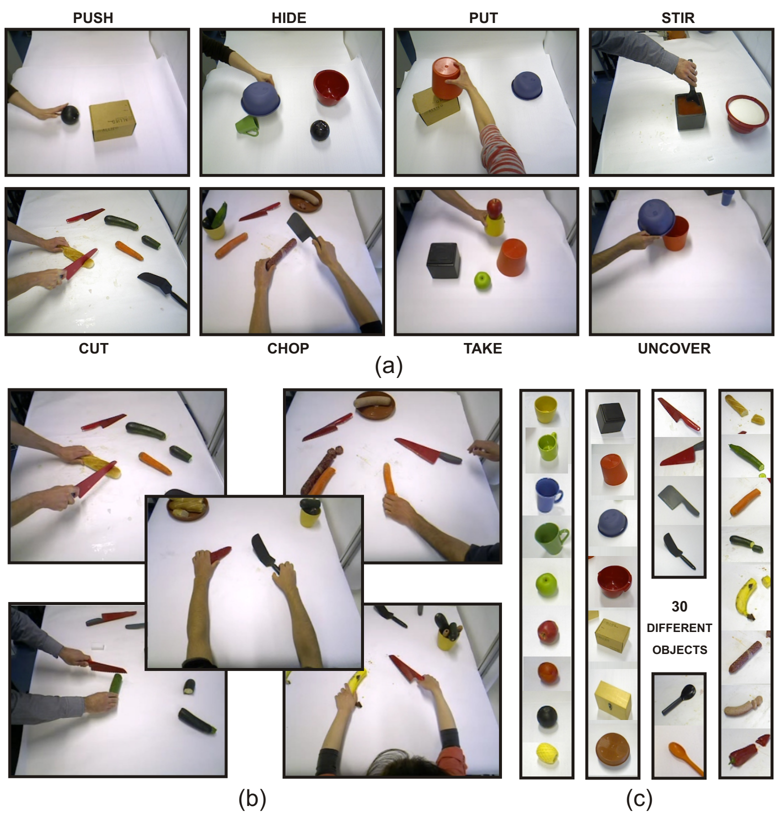


Figure 1: Eight different real action scenarios: *Pushing*, *Hiding*, *Putting*, *Stirring*, *Cutting*, *Chopping*, *Taking*, and *Uncovering*. (a) A sample original frame for each manipulation. (b) A sample frame from each demonstration of the *Cutting* action performed by 5 different individuals. (c) 30 different objects manipulated in all 120 manipulation demonstrations.

individual demonstration of the *Cutting* action to give an impression of the differences in demonstrations. There are in total 30 different objects manipulated in all 120 demonstrations. All manipulated objects are shown in Fig. 1 (c).

All manipulations were recorded with the Microsoft Kinect sensor which provides both color and depth image sequences. Colored objects are preferred to cope with the intrinsic limitations of the Kinect device. The central goal in these demonstrations is to learn a common archetypical SEC model for each manipulation including all possible variations in trajectory, pose, velocity, and object domains.

3.2. Segmentation and Tracking

The recorded image sequences are first pre-processed by a real-time image segmentation procedure to uniquely identify and track objects (including hands) in the scene. The segmentation algorithm is based on the color and depth information fed from the Kinect device and uses phase-based optical flow [24] to track segments between consecutive frames. Data transmission between these different pre-processing sub-units is achieved with the modular system architecture described in [25]. Segmentation and tracking approaches are described in detail elsewhere [26, 27], therefore, details are omitted here.

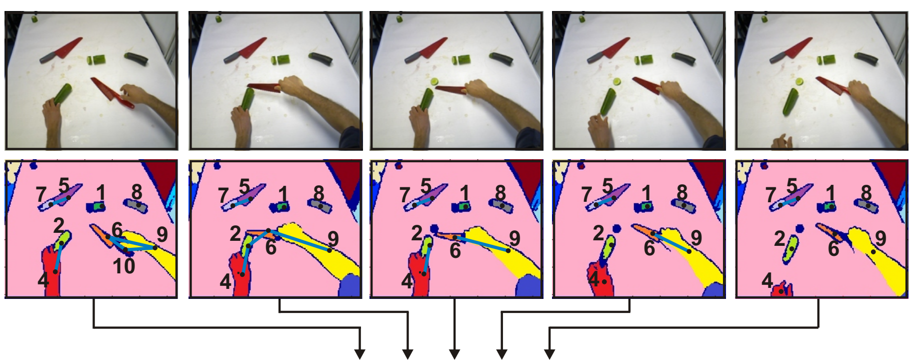
3.3. Extracting Semantic Event Chains (SECs)

Each consistently segmented image is represented by a graph: nodes represent segment centers and edges indicate whether two objects touch each other or not. By using the depth information we exclude the graph node for the background segment, i.e. supporting surface, since it does not play any crucial role in the dynamics of the manipulation. By using an exact graph matching technique, the framework discretizes the entire graph sequence into decisive main graphs. A new main graph is identified whenever a new node or edge is formed or an existing edge or node is deleted. Thus, each main graph represents a "key frame" in the manipulation sequence. All extracted main graphs form the core skeleton of the SEC, which is a matrix where rows are spatial relations (e. g.

touching) between object pairs and columns describe the scene configuration at the time point when a new main graph has occurred.

Fig. 2 depicts the SEC representation with some sample *key frames* including original images, respective segments (colored regions), and corresponding main graphs for one of the *Cutting* action demonstrations. For instance, the first row represents the spatial relations between graph nodes 9 and 6 which are hand and knife, respectively. Note that, although the whole demonstration sample has approximately 1000 frames, it is now represented by a 3×9 matrix.

Possible spatial relations are *Not touching (N)*, *Touching (T)*, and *Absence (A)*, where N means that there is no edge between two segments, i.e. graph nodes corresponding to two spatially separated objects, T represents objects that touch each other, and the absence of an object yields A. In the event chain representation, all pairs of objects need to be considered once, however, static rows which do not contain any change from N to T or vise versa are deleted as being irrelevant. For instance, the relation between the left and right



$$
\begin{array}{c}
9,6 \\ 4,2 \\ 6,2
\end{array}
\begin{bmatrix}
A & N & T & T & T & T & N & N & A \\
T & T & T & T & T & N & N & A & A \\
N & N & N & T & N & N & N & N & N
\end{bmatrix}
$$

Figure 2: SEC representation for a sample *Cutting* action where a hand is cutting a cucumber with a knife. Each column corresponds to one *key frame* some of which are shown on the top with original images, respective segments (colored regions), and main graphs. Rows are spatial relations between object pairs, e. g. between the hand (9) and knife (6) in the first row. Possible spatial relations are N, T, and A standing for *Not touching*, *Touching*, and *Absence*.

hand is always N and never switches to T to trigger an event, therefore, the respective row is ignored in the event chain. In Appendix A we introduce a de-noising process to cope with spurious spatial (rows) and/or temporal (columns) information propagated from noisy segmentation and tracking.

We note that there is no object recognition module included to identify graph nodes, i.e. segments, in the SEC framework. Event chains purely rely on spatial relational changes between segments in the temporal domain. The SEC extraction explained briefly in this section has been described in detail in [7].

3.4. Learning of Model SECs

The learning approach described next is an on-line unsupervised method to cluster observed SEC samples and to derive an archetypal SEC model for each cluster based on the semantic similarities between event chains. Each learned SEC model can then be used to describe a manipulation action.

Fig. 3 shows an overview of the proposed framework. The learning phase is triggered when a new manipulation experiment is observed; for example, a *Cutting* manipulation sample is introduced as the first experiment in Fig. 3. The new observed sample is represented by an event chain to be compared with the already learned SEC models. If there is no model existing, as in the case for this very first manipulation observation, the currently observed SEC sample N is directly assumed as a new model M_1. Once a new manipulation example is acquired, e.g. a *Chopping* sample as the second experiment in Fig. 3, the framework measures semantic similarities between the new SEC sample N and the known model M_1 in the spatiotemporal domain. We provide a detailed explanation of the similarity measure in Appendix B.

Semantic similarity values between the known models and the new sample are stored in a matrix, called the similarity matrix ($\zeta_{semantic}$), which is then converted into a histogram ($\mathcal{H}$) representing the distribution of similarities. We apply the conventional Otsu's method introduced in [28] to the normalized version of the histogram to further compute a threshold τ. See section 3.4.1 for the details of the derivation of $\mathcal{H}$ and τ from $\zeta_{semantic}$. The gray box in

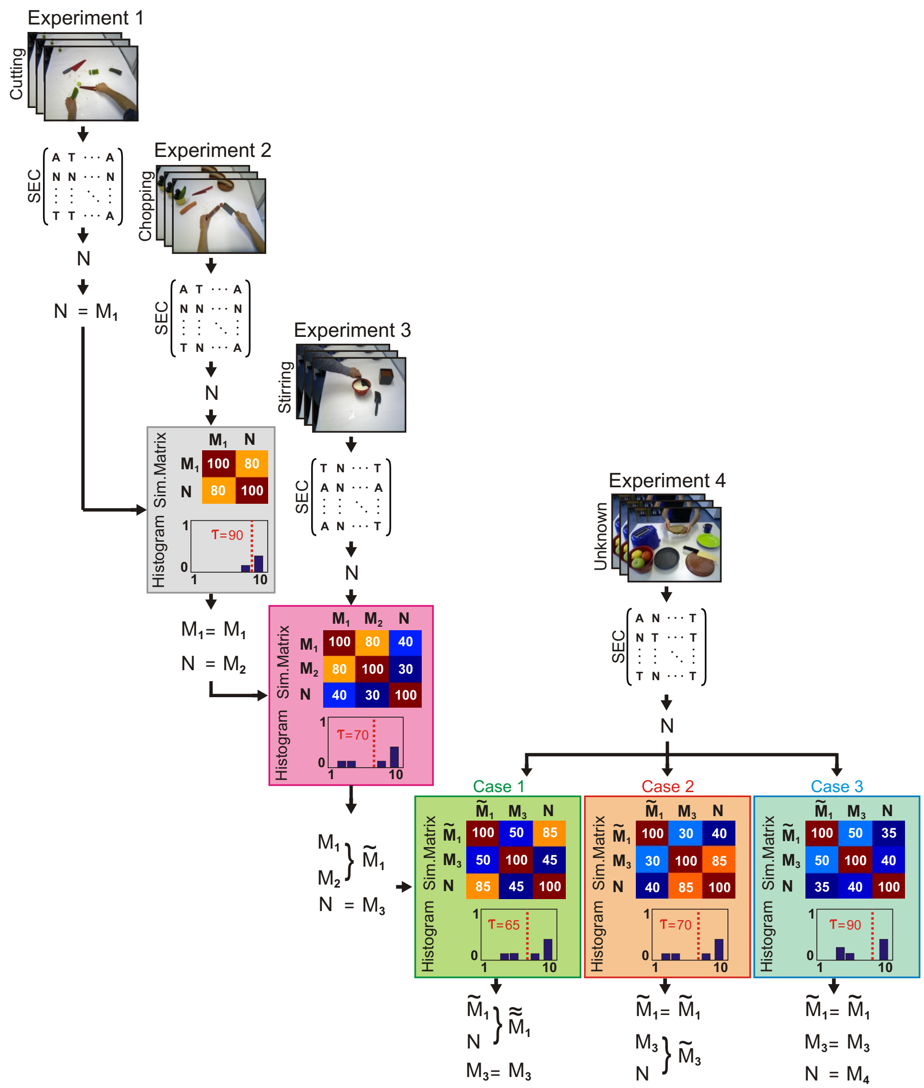


Figure 3: Overview of the proposed on-line learning framework.

Fig. 3 depicts extracted $\zeta_{semantic}$ and $\mathcal{H}$ in which the red dashed line indicates τ computed between the first two experiments.

Threshold τ is used for two purposes: First, we merge already learned SEC models which have higher semantic similarities than τ. Second, we compare the currently observed SEC sample with the so far existing models. If the comparison yields a higher similarity than τ, then the best fitting (highest similar) model will be refined with the new SEC sample. Otherwise, a new model will be created based on the SEC sample.

The comparison of the first two experiments N and M_1 shown in the gray box in Fig. 3 yields 80% semantic similarity which is less than τ estimated as 90% (See Appendix B). Therefore, the *Chopping* sample N is considered as a new SEC model M_2. We repeat the same procedure, i.e. computing $\zeta_{semantic}$, $\mathcal{H}$, and τ, once the next sample N, which is a *Stirring* experiment in this case, is observed. As depicted in the purple box τ drops below 80% which allows us to update M_1 with M_2 yielding $\widetilde{M_1}$. As the *Stirring* demonstration still has less similarities with any of the known models, a new model M_3 is initialized with N.

The threshold value is required to better assess the obtained semantic similarities between models and the observed sample. Therefore, whenever a new observation is available, the entire process of estimating a new τ by determining $\zeta_{semantic}$ and $\mathcal{H}$ is repeated to decide on the fly whether the current SEC sample belongs to one of the already learned manipulation models or whether it represents a new manipulation. This is summarized with the fourth experiment introduced as an *Unknown* demonstration in Fig. 3, the fate of which depends on three possible cases. Case 1 and 2 are respectively standing for the processes of refining the models $\widetilde{M_1}$ and M_3 with N, whereas Case 3 is representing the initialization of a new model M_4.

In the following, we will describe how to compute the threshold and update a learned model with a new SEC sample.

3.4.1. Computing the Threshold

Let $\mathcal{M}$ be a set of learned SEC models at any observation time as

$$\mathcal{M} = \{m_1, m_2, \cdots, m_n\} \quad , \tag{1}$$

where n is the total number of existing models. Semantic similarity values between all learned models are stored in a matrix as

$$\zeta_{semantic} = \begin{bmatrix} \varphi_{1,1} & \varphi_{1,2} & \cdots & \varphi_{1,n} \\ \varphi_{2,1} & \varphi_{2,2} & \cdots & \varphi_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{n,1} & \varphi_{n,2} & \cdots & \varphi_{n,n} \end{bmatrix} \quad , \quad 0 \leq \varphi_{i,j} \leq 100 \quad and \quad \varphi_{i,j} = \varphi_{j,i} \quad ,$$

where $\varphi_{i,j}$ holds the semantic similarity between models m_i and m_j and is computed as described in Appendix B.

Semantic similarity matrix $\zeta_{semantic}$ is then converted into a histogram $\mathcal{H}$ representing the distribution of similarities as

$$\mathcal{H} = \{h_k : \ k \in [1, \cdots, \lambda]\} \quad , \tag{2}$$

$$h_k = \frac{1}{\eta} \sum_{i=1}^{n} \sum_{j=i}^{n} \delta_{i,j} \quad , \tag{3}$$

$$\delta_{i,j} = \begin{cases} 1 & \text{if} \ \ \frac{\varphi_{i,j}}{\phi} \ \ \text{is at bin } k \\ 0 & \text{else} \end{cases} \quad , \tag{4}$$

where λ is the total number of bins each has a size of ϕ which is chosen as 10 in our experiments and η is the normalization factor. Note that, since the similarity matrix $\zeta_{semantic}$ is symmetric, only half of the matrix is processed, thus, the value of j changes from i to n in Eq. (3) and η is defined as $n(n+1)/2$.

The normalized histogram $\mathcal{H}$ is now used to calculate the required threshold using the conventional Otsu's method introduced in [28]. For this purpose, we compute zero- and first-order cumulative moments of the normalized histogram at each bin as

$$\omega(k) = \sum_{i=1}^{k} h_i \quad , \qquad (5) \qquad\qquad \mu(k) = \sum_{i=1}^{k} i h_i \quad . \qquad (6)$$

The total mean value of the histogram is calculated as

$$\mu_T = \sum_{i=1}^{\lambda} i h_i \quad . \tag{7}$$

The variance of the histogram separability is then given by

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \quad . \tag{8}$$

Otsu's method yields a threshold value k^* for that bin at which the variance σ_B^2 is maximal; that is,

$$k^* = \underset{1 \leq k < \lambda}{\arg\max} \, (\sigma_B^2(k)) \quad . \tag{9}$$

The threshold k^* separates the histogram into two distinct regions. The left side of k^* indicates low semantic similarity between models in $\mathcal{M}$, and vice versa. As we are seeking for a threshold τ to group similar manipulations, we take the average of the similarities falling into the right side of k^* as

$$\tau = \frac{1}{\eta_r} \sum_{i=k^*}^{\lambda} h_i \quad , \tag{10}$$

where η_r is the normalization term which is the total number of similarity values on the right side of k^*.

3.4.2. Updating Model SECs

Once the highest semantic similarity between a novel SEC sample and any of the known models is higher than the threshold τ, the one model with *highest* similarity to the new SEC is now updated with this new SEC sample. To update a model, the learning procedure just needs to search for all common rows and columns observed in the new SEC sample.

Each model is initially created by assigning weight value of 1 for each row. Once a new SEC sample is observed, weights of each row in the model that match to a row in the new SEC are incremented. This way existing common rows between the matched model and the novel sample are receiving increasing weights. In the case of having additional rows in the new SEC sample, the model is extended by these rows, each of which is initiated again by giving them a weight of one. As the next step, we search for the common temporal information embedded in the columns of the event chains by employing a procedure very similar to that applied for extracting common rows. Finally, the model SEC consists of only those rows and columns observed frequently in the observed new SEC samples. A detailed explanation of the model updating procedure is given in Appendix C.

4. Results

In this section, we will first show experimental results from our proposed incremental learning framework. We will then continue with enrichment of each learned SEC model with object information. Next, validation and testing processes of the learned models will be given.

4.1. Learning

We apply the incremental learning and clustering framework described above to 8 different manipulation actions each of which has 15 versions, yielding in total 120 samples, as introduced in section 3.1. Manipulation tasks have vast variations in terms of manipulated objects, their poses, and followed trajectories as depicted in Fig. 1. The framework is first tracking each segment in the scene and extracting the corresponding SEC representation from a randomly observed manipulation sample. While observing more samples, different SEC models are learned or updated based on the threshold value.

When we let the framework run only once through 120 manipulation tasks by randomly choosing a sample at each time, it learns 22 model event chains.

Fig. 4 (a) shows the final computed semantic similarity matrix $\zeta_{semantic}$ between each of the learned models. Low similarities between models indicate how distinct those models are. The corresponding histogram representation $\mathcal{H}$ with derived thresholds k^* and τ is depicted in Fig. 4 (b). The threshold k^* separates the histogram into two distinct regions as depicted with the gray shade and τ is then calculated as 72 from Eq. (10). In Fig. 4 (c), we can see the complete behavior of τ during the learning cycle with 120 observation samples. It is initiated with 100 and after updating with Eq. (10) at each observation it starts to converge to approximately 72.

Fig. 5 (a) depicts all learned models with corresponding number of observation samples employed for updating each. The green dashed line indicates the actual sample numbers as the ground truth. Although the framework learns in total 22 models, only 7 of them, those in the red box, contain more than 10 samples and the rest hold at most 2 samples. Recalling the fact that the training set has 8 manipulations, we can state as one central result that 7 of them are indeed found with high numbers of examples each. *Cutting* and *Chopping* models are merged, though, but we will explain below that this actually "makes more sense" than the naively (by us) assumed ground truth. Furthermore, we

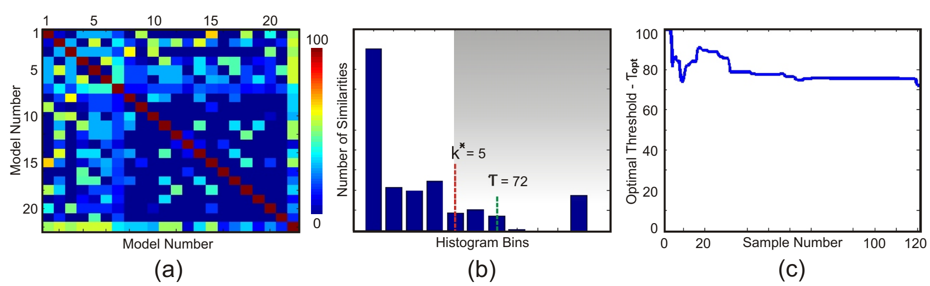


Figure 4: Thresholding. (a) Semantic similarity matrix $\zeta_{semantic}$ computed between 22 learned SEC models. The scale bar on the right indicates the similarity values in percent. (b) Respective histogram representation $\mathcal{H}$ with extracted k^* and τ values. The threshold k^* separates the histogram into two distinct regions as depicted with the gray shade. (c) Development of τ during the observation of 120 samples. τ is initiated with 100 and after updating with Eq. (10) it starts to converge to 72.

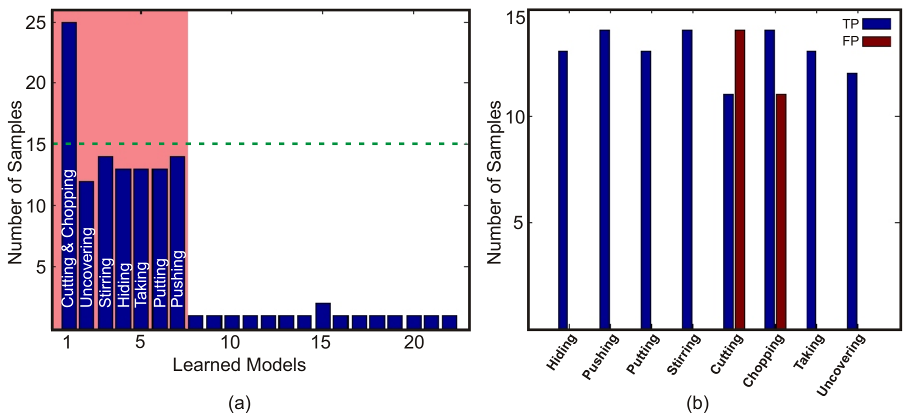


Figure 5: Number of learned models and clustering accuracy of observed samples. (a) Learned 22 SEC models with corresponding number of trained samples. The green dashed line indicates the actual sample numbers as the ground truth. (b) Number of true and false positive samples clustered in learned models with respect to the ground truth.

observe that only few demonstrated samples have either enormous variations or noise, i.e. less semantic similarities than τ with any other models, which leads to the generation of the additional models outside the red box. As mentioned, our framework produces a single model representing the *Cutting* and *Chopping* manipulations together due to having high semantic similarities. It is because both manipulations have the same fundamental action primitives, i.e. similar columns in the event chains, and the only differences are mostly in the followed trajectories and velocity of the movements which are not captured by SECs. See Fig. B.14 in Appendix B as an example of high semantic similarities between the *Cutting* and *Chopping* tasks. Thus, Fig. 5 (a) shows that without using any human intervention the proposed learning framework can automatically retrieve the demonstrated 8 manipulation types two of which are naturally merged.

As addressed in section 3.4, all manipulation samples used for updating the same SEC model will have the same cluster label. In Fig. 5 (b), we show the number of true and false positive samples falling into the same model with respect to the ground truth. Except for the *Cutting* and *Chopping* manipulations, none of the given manipulation samples is wrongly clustered. This means, for instance, a given *Stirring* demonstration is used only for updating the *Stirring* model, but not for the *Pushing* model, etc. However, since we have now only one SEC model for the *Cutting* and *Chopping* manipulations, samples from both manipulations will be used for the same model. As the ground truth expects two different models, high false positives are observed for both.

Fig. 6 shows how the clustering results for all 120 manipulation samples are varying from observation to observation. Colors encode the cluster labels and the ground truth for each cluster is given on the left. Note that time is progressing from left to right, thus the first observed sample is the one depicted in cyan in the *Chopping* manipulation. As a consequence of merging models with high semantic similarity, some clusters will merge once new observations become available. Black ellipses depict when a sample switches from one cluster to another. For instance, cyan clusters observed for the *Chopping* samples in the

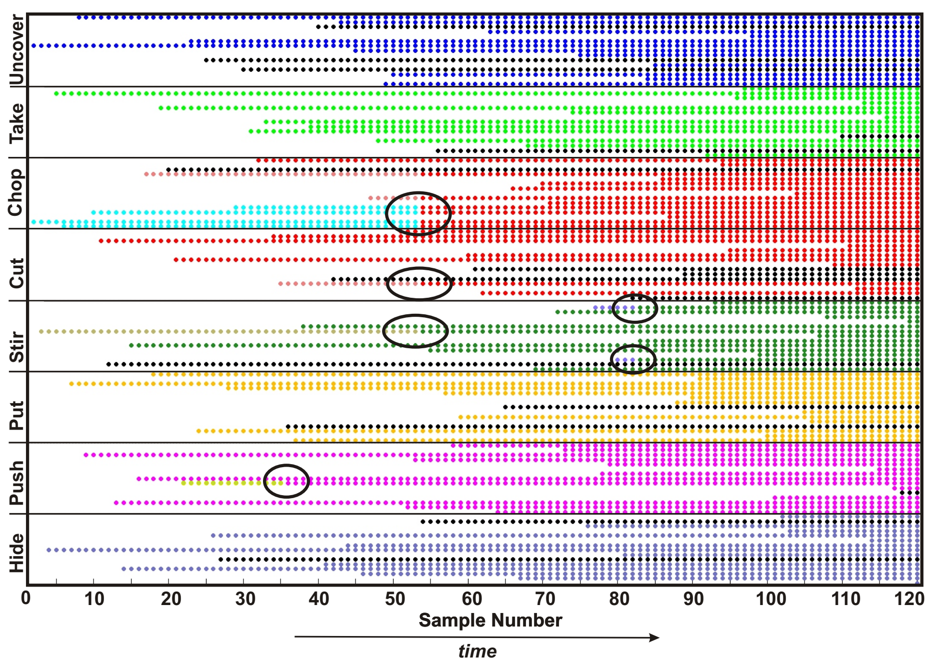


Figure 6: Clustering result of 120 manipulation samples. Colors encode the cluster labels and the ground truth for each cluster is given on the left. Noisy clusters are indicated in black. Black ellipses depict when a sample switches from one cluster to another.

beginning are turned into red clusters originally created for the *Cutting* task. At the sample number 120 in the very right hand side we therefore observe 7 different colored clusters each from one learned model. This figure illustrates that for some manipulations types the model is immediately converging to the optimal solution, whereas for other models certain number of samples are required. Noisy clusters, which belong to the noisy models shown outside the red box in Fig. 5 (a), are indicated by black dots.

To investigate the robustness of the framework, we repeat the same learning experiment explained above 100 times independently from each other and compute differences between the learned models. In each trial, the framework produces at least 21 and at most 23 various models. However, when we compare all these models extracted in 100 trials, we see indeed 29 different ones, the distribution of which is shown in Fig. 7 (a). Among those 29 models, it is again the same 7 models introduced in Fig. 5 (a) which have high number of samples. Furthermore, as indicated in Fig. 7 (b) we still do not obtain any false positives among the clustered samples except for the *Cutting* and *Chopping* manipulations due to the same reason as clarified above. Note that the red bars depict the standard error of the mean for those which are not zero. Fig. 7 consequently

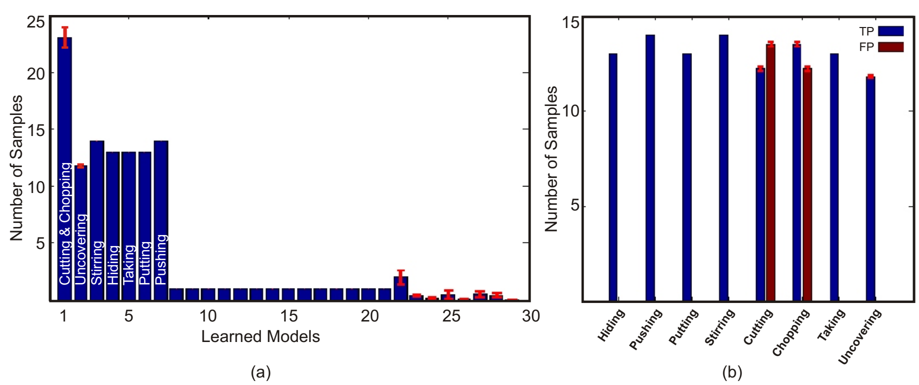


Figure 7: Total number of learned models and clustering accuracy after 100 independent trials. (a) Learned 29 SEC models with corresponding number of trained samples. (b) Number of true and false positive samples clustered in learned models with respect to the ground truth. Red bars depict standard error of the mean for those which are not zero.

proves that the learning approach is always converging to the same 7 models no matter in which order the manipulation samples are provided.

We can now take a close look at some of those 7 SEC models explored from demonstrated manipulation actions. Fig. 8 shows models for the *Cutting* & *Chopping*, *Stirring*, and *Uncovering* manipulations with all derived states introduced in Eq. (C.2) and the transition probabilities between each. States and arrows given in red color correspond to the most commonly observed event chain columns and their transitions with the highest probabilities as described in Eqs. (C.3) and (C.4), respectively. On the left side of each model, we also show weight values ($\mathcal{W}$ from Eq. (C.1)) for each row in the states. It can be

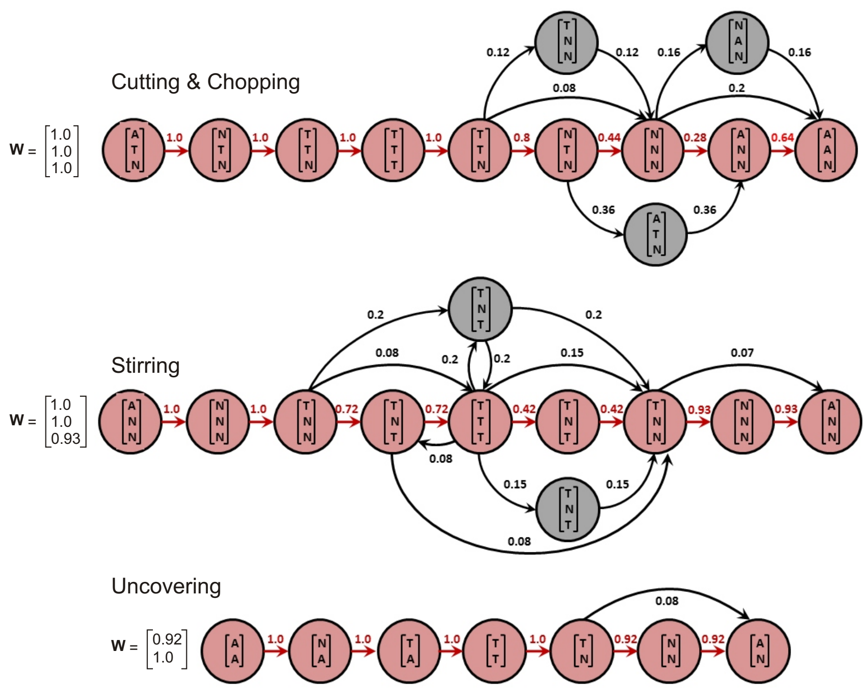


Figure 8: Complete learned SEC models for the *Cutting* & *Chopping*, *Stirring*, and *Uncovering* manipulations. Each state corresponds to one SEC column and arrows represent the transition probabilities from one state to the next. Those in red color correspond to the most commonly observed states, their transitions having the highest probabilities. Weight values $\mathcal{W}$ on the left indicate how often each row in the states is obtained in the trained samples.

seen that in all 3 models some rows are quite commonly obtained in the trained samples since their weights are close to 1, whereas this is not the same for the state transitions. For instance, in the *Cutting* & *Chopping* model, there exist three more states given in gray color which are particularly observed in the second half of the action and cause drop of some state transitions to 0.28. This is because even though each subject grasps a tool and cuts or chops an object in the same temporal order, they leave the scene in different orders; for example, one subject first removes the hand supporting the object to be cut and then withdraws the hand holding the cutting tool whereas another subject either does it the other way around or removes both hands at the same time. Another reason of having extra states, thus smaller transition probabilities, is the noise propagated from the segmentation and tracking components as observed in the *Stirring* model. Nevertheless, we can now extract all these variations that occurred due to the nature of manipulation or noise and pick the most often observed states, i.e. states in red, as a representative model for each manipulation action. Note that the learning process never ends and is open to refine models incrementally whenever new samples are provided, just like the assimilation process that happens in humans [29].

4.2. Enriching Learned Model SECs

In this section, we will show how learned SEC models can be enriched with additional object information.

During the updating process of model SECs, we determine correspondences between rows of event chains as explained in section 3.4.2. Since each row in an event chain holds relational changes between segments in the scene, the row correspondences can also be used to calculate matchings between segments in two event chains. We refer the interested reader to [7] for details of the segment matching method. We now use this technique to extract segments, i.e. objects, that play the same role in different versions of the same manipulations observed during the learning phase.

Fig. 9 (a) shows the learned *Cutting* & *Chopping* model, columns of which are the states indicated in red in Fig. 8. The framework now estimates which segment is used as a main tool and which one as an object to be cut or chopped in each observation. As explained in Appendix A, we refer to the hand as the *manipulator* and to the object interacting with the hand as the *primary object*, e.g. a knife or a cleaver. Other objects which are combined with the primary object are called *secondary objects* like the cucumber to be cut. Note that the second hand is almost always used to help the *manipulator*, hence it is called the *supporter*. Fig. 9 (a) illustrates all matched *primary* and *secondary objects* used for training of the *Cutting* & *Chopping* model. Fig. 9 (b) shows the *primary* and *secondary objects* for the *Stirring* model. In this case, not only a spoon but a knife and a spatula are also selected by subjects as the *primary object* used for stirring. The *secondary object* is the stirred liquid and the buckets are the *supporters*. As learned model SECs are refined with every new observation, all these variations of the different objects will be attached to the model, simultaneously. Note that segments representing the *manipulator* and *supporter* are also matched, however, are not shown due to lack of space.

It is important to underline that the proposed framework is not utilizing any object recognition method, hence, we are here strictly at the level of segments. For the sake of simplicity, object images are shown instead of segments in Fig. 9. It is evident that this unsupervised segment categorization process could be coupled to object models, thus, providing access to object categorization, too.

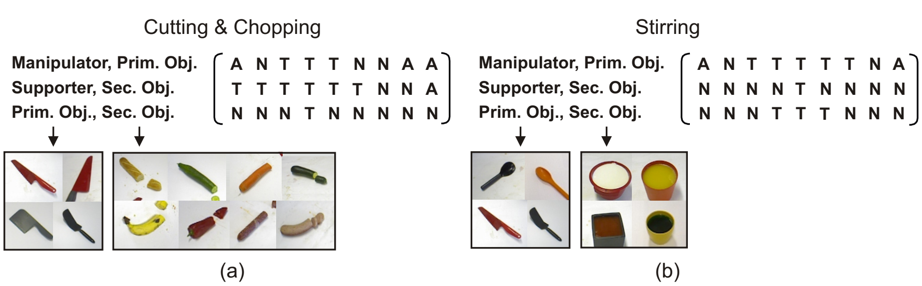


Figure 9: Learned *Cutting* & *Chopping* and *Stirring* models enriched with object information. Each column in the SEC model corresponds to one state indicated in red in Fig. 8. *Primary* and *secondary* objects are extracted from observed manipulations during the learning process.

4.3. Validation and Testing

A validation process of the learned 7 SEC models is performed with the classification of all 120 training samples according to their semantic similarities with the learned models. This step is required to show the clustering accuracy of the training data but nothing unexpected will be observed here. We note that the main and critical evaluation is then shown by the next following testing experiment with a set of novel and complex manipulation *sequences*.

We label each SEC model as a different class and introduce a static threshold chosen as 72 which is the converging value (τ) obtained during the learning phase as depicted in Fig. 4 (c). Once the highest semantic similarity between a training sample and any of the known models is higher than this threshold, the sample is assigned to that class. The classification method has also a class type called *Unknown* to detect samples that have low similarities with all known models.

Fig. 10 (a) shows the confusion matrix depicting the classification accuracies of the complete training data set with respect to the learned models. The first impression that the figure conveys is that there is no misclassification of any training data; for instance, 87% of the *Hiding* training manipulations are

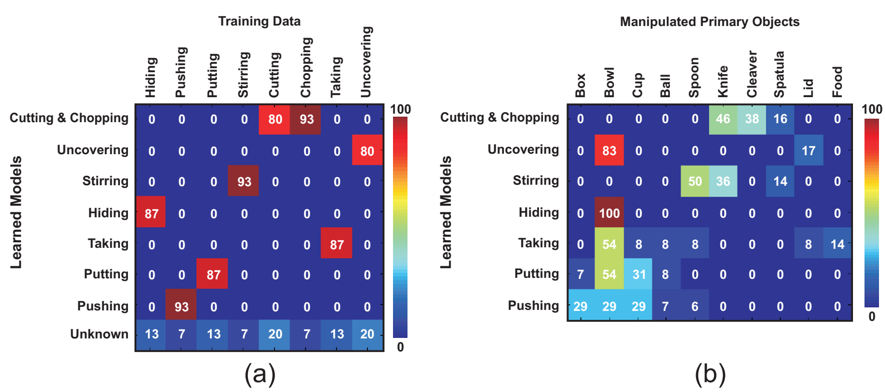


Figure 10: Confusion matrix showing (a) the classification accuracy for the complete training data set including in total 120 samples and (b) the usage rate of different objects primarily manipulated in the learned models.

correctly classified in the model *Hiding* and the rest is assigned as *Unknown*. As there is only one representative SEC model existing for both *Cutting* and *Chopping* manipulations, training samples from those are assigned within the same model *Cutting* & *Chopping*. The validation phase of the complete training set leads to 100% average precision and 87% average recall.

As addressed in section 4.2, we can also extract objects which are manipulated in a similar manner in different demonstrations of the same manipulation type. Fig. 10 (b) indicates the primary object types frequently manipulated in each classified training data. It is observed that objects like *Knife*, *Cleaver*, and *Spatula* are manipulated often in the *Cutting* & *Chopping* model class, whereas, due to its size, *Bowl* is the only preferred object in the *Hiding* manipulation to cover other objects. Fig. 10 consequently proves the high success rates of the discriminative and descriptive features of the learned 7 SEC models and their direct relations with manipulated objects.

To further evaluate the performance of the learned model SECs, we create a new testing set with 20 long chained actions which consist of in total 103 different versions of the learned single manipulations such as *Cutting*, *Stirring*, and *Pushing*. We also introduce a new manipulation type called *Pouring* to measure the responses of the learned SEC models against a novel manipulation. In each chained action the subject has a certain task, e.g. "*making a sandwich*", which involves execution of multiple single manipulations in various orders, either sequentially or parallelly. Fig. 11 depicts sample frames from two different chained action sequences in which subjects are performing the same task "*making a sandwich*" by using novel objects in various ways to increase the complexity of the scenes. We here apply an unsupervised, probabilistic method that measures the frequency of the changes in the spatial relations embedded in event chains to extract the main manipulator, e.g. hand, and to decompose the long chained actions into their primitive action components according to the spatiotemporal relations of the manipulator. Hence, also the decomposition process is model free and automatic. Since the decomposition issue is not in the core of the proposed framework, we omit the details here and refer the interested

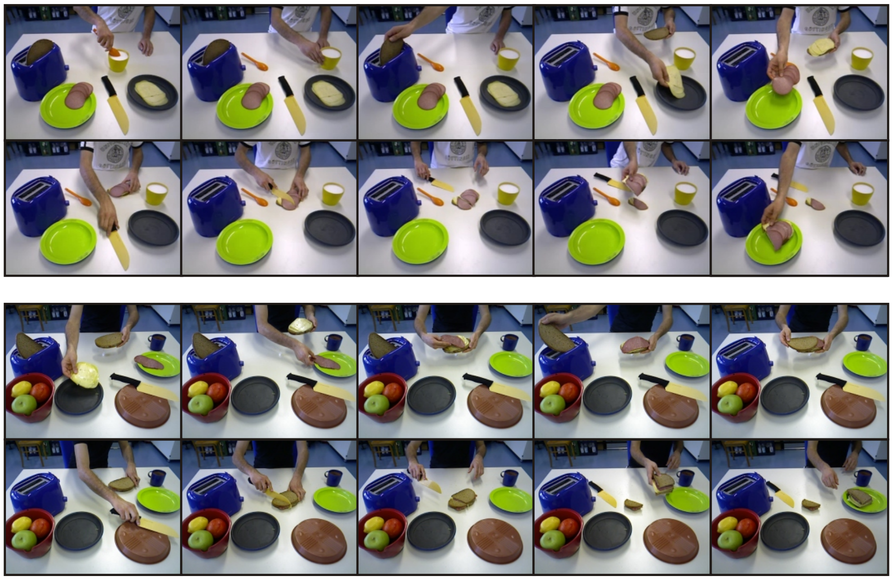


Figure 11: Sample frames from two different long chained manipulation sequences which are used to test the learned SEC models. In these demonstrations subjects are performing the same task "*making a sandwich*" by using novel objects in various ways to increase the complexity of the scenes.

reader to [30].

Each single decomposed manipulation action is again analyzed as a classification task as described in the validation phase. Fig. 12 (a) indicates the highly successful classification results of decomposed manipulations with respect to the learned models. We receive minimum 83% accurate classification rate which is for the *Stirring* manipulation and maximum 10% misclassification rate as observed only for the *Pushing* manipulation. It is also significant to note that the novel *Pouring* manipulation is never confused with any of the known SEC models. In this testing phase, average precision and recall values are measured as 99% and 96%, respectively.

Fig. 12 (b) shows the most often manipulated primary object types in each classified test data. Compared to the results obtained in the training phase, the major difference here is the high usage rates of the object type *Food* in the *Hiding*, *Taking*, and *Putting* models. This is because making a sandwich by

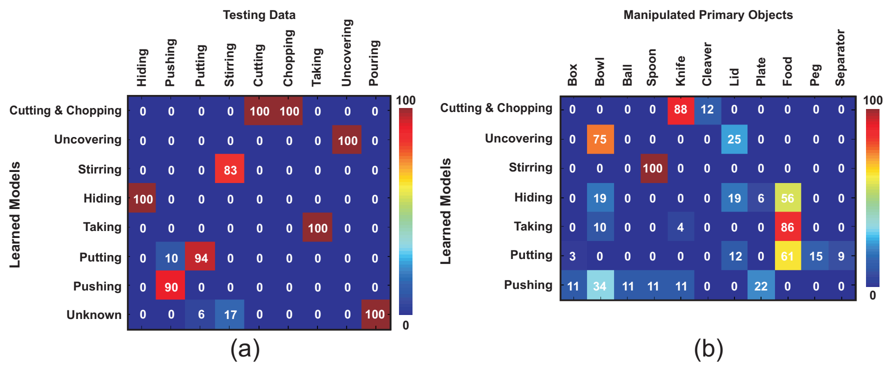


Figure 12: Confusion matrix showing (a) the classification accuracy for the complete testing data set including in total 103 samples and (b) the usage rate of different objects primarily manipulated in the learned models.

taking and putting cheese or bread slices on top of each other naturally results in occlusions as expected by the *Hiding* model.

Note that all results shown in Figs. 10 and 12 are acquired in a fully automated, unsupervised manner and show that the learned SEC models are highly accurate and discriminative to recognize manipulation actions which can even be embedded in the long and complex chained demonstrations performed with novel objects under different circumstances.

5. Discussion

The main contribution of our paper is a novel method for incrementally learning the semantics of manipulation actions by observation. The proposed learning framework is bootstrapped with the semantic relations (SECs) between observed manipulations without using any prior knowledge about actions or objects while being fully grounded at the sensory level (image segments). To our best knowledge this is one of the first attempts in cognitive robotics to infer descriptive semantic models of observed manipulations in a fully automated and unsupervised manner.

One of the most fundamental advantages of the proposed framework is that during the learning, when a new sample is observed, it is not compared with all previously acquired samples, which is an exhausting operation, but instead is compared only with the already learned models which are then updated accordingly. This is of importance to allow the cognitive agent to use its memory in a more efficient way for lifelong learning, which is known as "Assimilation Process" in human cognition as originally defined by Piaget [29].

The proposed framework can be easily enriched with object information since event chains naturally group objects considering only their performed roles in a manipulation. As a strong contribution, we showed that objects, i.e. segments, can be categorized based on how an object is being manipulated, rather than by knowing what type of object it is. As shown in our previous works [9, 23], not only object but pose and the followed trajectory information can also be embedded into the SEC representations as further enrichment.

In this paper, we also introduced a new manipulation action data set with 8 different manipulation tasks (e.g. *Cutting*, *Chopping*, *Stirring*, etc.), each of which consists of 15 different versions performed by 3 different human actors. This data set was used to learn an archetypal SEC model for each manipulation action. To further quantitatively evaluate the learned SEC models, we extended our data set with 20 long and complex chained manipulation sequences (e.g. "*making a sandwich*" or "*preparing a breakfast*") which consist of in total 103 different versions of these 8 manipulation tasks performed in different orders with novel objects under different circumstances. These data sets are publicly available and could be used for action/object benchmarking also of other methods.

In contrast to other well-known data sets, our new benchmark set captures manipulation activities from the subjects's own point of view with a static RGB-D camera since we are interested in understanding the spatiotemporal interactions between the manipulated objects and hands. The conventional data sets, however, employ the entire human body configurations and movements as main features and therefore either do not involve hand-tool features [31, 17, 32] or are not rich to provide enough recordings required for the learning [15, 16].

The observed high accuracy of our method when classifying the unknown (long-sequence) test-data set support that the learned models are indeed discriminative and descriptive of these actions (and objects). The here shown experimental results also exhibit a similar behavior to that of the ontologies presented in [33, 15]. In these both studies manipulation actions were classified into six distinct structural categories (e.g. *Rearrange*, *Destroy*, *Break*, etc.) in which *Cutting* and *Chopping* manipulations were subsumed in the same category as the learned single Cutting & Chopping model in our framework.

As mentioned in the introduction, additional information, if available about a given action, will further improve action understanding. This notwithstanding, we believe that the current study strongly supports the power of the Semantic Event Chain framework, because here we have "pushed it to an extreme" by fully relying on model-free, unsupervised algorithms for clustering and classification. Therefore, we would hope that this study might stimulate the research community to adopt this framework in the future.

Appendices

We here provide three appendices each describes details of individual algorithmic steps in details. The first appendix introduces the de-noising process to filter out noisy spatiotemporal relations in the event chains. In the next appendix, the detailed description of the similarity measure between event chains is given. The last appendix highlights the updating process of a learned SEC model with a novel SEC sample.

Appendix A. De-noising of SECs

Due to some early vision problems such as illumination variations or occlusions observed in the segmentation and tracking phases, extracted event chains can contain noisy spatial (rows) and/or temporal (columns) information. To prevent noisy event chain elements to propagate further to the next learning stage, we apply a de-noising process to the extracted raw SECs. The de-noising process is based on reasonable action descriptive assumptions (rules) introduced in [33], which are as follows:

1. only *single hand* manipulations are considered;
2. the hand can manipulate, i.e. *touch*, only one object at a time;
3. the manipulation can take place at the touched object itself (the one mentioned in rule 2) or only one other object can be a target, with which the first one interacts, i.e. *touches*;
4. before and after the manipulation the hand is *free* and *not-touching* anything;
5. before and after the manipulation the hand is *not in the scene*.

The first two rules guarantee that there is only one hand and at most one object interacting with the hand, which we call *manipulator* and *primary object*, respectively. Other objects, which are combined with the primary object, are called *secondary objects*. The third rule assures that *manipulator*, *primary* and *secondary objects* are the only ones having direct interaction with each other affecting the dynamics of the manipulation. The last two rules define the natural start and end points of the manipulation.

The de-noising process checks whether all those rules are satisfied in the SEC representation. For instance, the first two rules require that the event chain must have a row holding spatial relations between the *manipulator* and *primary object* and last three rules define these relations as:

$$\textit{manipulator}, \textit{primary object} \quad \begin{bmatrix} A & N & T & \cdots & T & N & A \end{bmatrix}, \tag{A.1}$$

where the *manipulator* is first absent (A) in the scene *(rule 5)*, then appears but does not touch (N) the *primary object (rule 4)*. Next, the *manipulator* touches (T) *primary object* to apply a certain task on it *(rule 3)*. Depending on the manipulation, the temporal length of the touching (T) relation can vary. Finally, the *manipulator* releases (N) the *primary object (rule 4)* and leaves (A) the scene *(rule 5)*.

Since segments, i.e. graph nodes, are not identified as objects in event chains, we do not know which segment corresponds to the *manipulator* or *primary* *object*. Therefore, we apply a probabilistic reasoning to estimate segment roles in the manipulation. Probability values for each segment are assigned based on similarities of their relations with Eq. (A.1) and the frequency of their touching relations. See Appendix B for similarity calculation between SEC rows. In this regard, all rows in the event chain are compared with Eq. (A.1) and the most similar one is taken as the best candidate for the *manipulator* and the *primary object*.

Fig. A.13 (a-b) shows a noisy raw event chain with corresponding *key frames* extracted from a *Putting* manipulation sample where a hand is putting a cup on a box. For instance, the first and second rows of the SEC given in Fig. A.13 (b) are similar to Eq. (A.1), however, the second row has a higher probability to be a better candidate due to having more touching relations. Therefore, segments 4 and 1 in the second row have the highest likelihood to be the *manipulator*

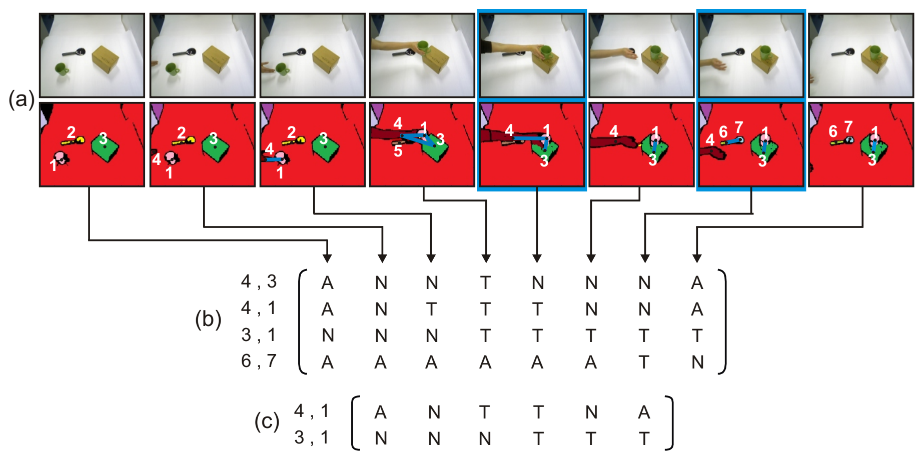


Figure A.13: SEC representation for a sample *Putting* action where a hand is putting a cup on a box. (a) Extracted 8 *key frames* with original images, corresponding segments (colored regions), and main graphs. (b) Respective SEC where each *key frame* corresponds to one column. Rows are spatial relations between object pairs, e. g. between the hand (4) and box (3) in the first raw. Possible spatial relations are N, T, and A standing for *Not touching*, *Touching*, and *Absence*, respectively. (c) De-noised SEC after applying action descriptive rules. First and last rows as well as repetitive key frames, shown in blue frames, are removed from the raw SEC in (b).

and the *primary object*. Since *rule 5* constrains the *manipulator* to appear in the scene later, we choose segment 4 as the *manipulator* and segment 1 as the *primary object*.

Once the *manipulator* and *primary object* are estimated, the de-noising process is concluded by examining the second and third rules once more. Since the second rule does not allow the *manipulator* to interact with any other object other than the *primary object*, such rows can be considered as noise to be omitted. In this manner, the first row in the SEC given in Fig. A.13 (b) can be ignored as the *manipulator* (segment number 4) is also touching the box (segment number 3) which is not the *primary object*. Note that the hand is here accidentally touching the box while putting the cup. Recalling the third rule, we can ignore any segment which does not have any interaction with the *manipulator* or *primary object*. In this sense, the forth row of the SEC in Fig. A.13 (b) is omitted because segment 6 and 7 represent the spoon which is occluded by the *manipulator* and *primary object* and not playing any role in the manipulation. Fig. A.13 (c) shows the final de-noised SEC representation for the *Putting* action in Fig. A.13 (a). Note that de-noised event chain includes less columns since redundant duplicate (repetitive) columns observed after deleting noisy rows (indicated in blue frames in Fig. A.13 (a)) are also removed.

It is important to underline that the de-noising process considers temporal interactions between entire segments in the manipulation to solve illumination or occlusion based early vision problems which can not be solved without reasoning at a higher level.

Appendix B. Measuring Semantic Similarity

Once event chains are extracted in the observation phase, their semantic similarities need to be compared to further explore whether they describe the same type of manipulation. In [7], we introduced a method to measure semantic similarities and here we describe an updated version which is more robust against noisy spatiotemporal information coming from the early vision stage. To better explain the semantic comparison we will use sample demonstrations from the *Cutting* and *Chopping* manipulations which are shown in Fig. B.14 (a-b) with extracted de-noised SECs including some sample key frames with respective segments and graphs. Note that even though those two samples have different perspectives and contain different number and types of objects, the dimensions of the event chains are accidentally the same. This is of no importance as our proposed method does not rely on dimensions, allowing to compare arbitrarily long manipulations.

To calculate the semantic similarity between two manipulations, spatial and temporal aspects are being analyzed in two separate steps. In the first step, we compare spatial information, i.e. relational changes in each row, and in the following second step the temporal information, i.e. the order of columns, is considered. In both steps we apply a standard sub-string search algorithm. To achieve this, we first perform a data-compression on the original chain (ξ_o) by simply scanning each row of ξ_o from left to right and substitute "changes" by combining their values into a two-digit format. For example a change from *Not touching* to *Touching*, hence from N to T, is now encoded by NT. When nothing has changed, a double digit like TT, is removed. This compressed event chain, represented by ξ_c, lost all temporal information and is used only for the spatial-relational analysis in the first step. The original chain (ξ_o) will then be used for the temporal analysis in the second step. ξ_o and ξ_c of the *Cutting* and *Chopping* actions are given in Fig. B.14 (a-d).

Let ξ_c^1 and ξ_c^2 be the sets of rows for the two manipulations, written as a matrix (e.g. Fig. B.14 (c) and B.14 (d)):

$$\xi_c^1 = \begin{bmatrix} r_{1,1}^1 & r_{1,2}^1 & \cdots & \cdots & r_{1,\gamma_1^1}^1 & & \\ r_{2,1}^1 & r_{2,2}^1 & \cdots & r_{2,\gamma_2^1}^1 & & & \\ \vdots & \vdots & \ddots & \vdots & & & \\ r_{m,1}^1 & r_{m,2}^1 & \cdots & \cdots & & \cdots & r_{m,\gamma_m^1}^1 \end{bmatrix},$$

and

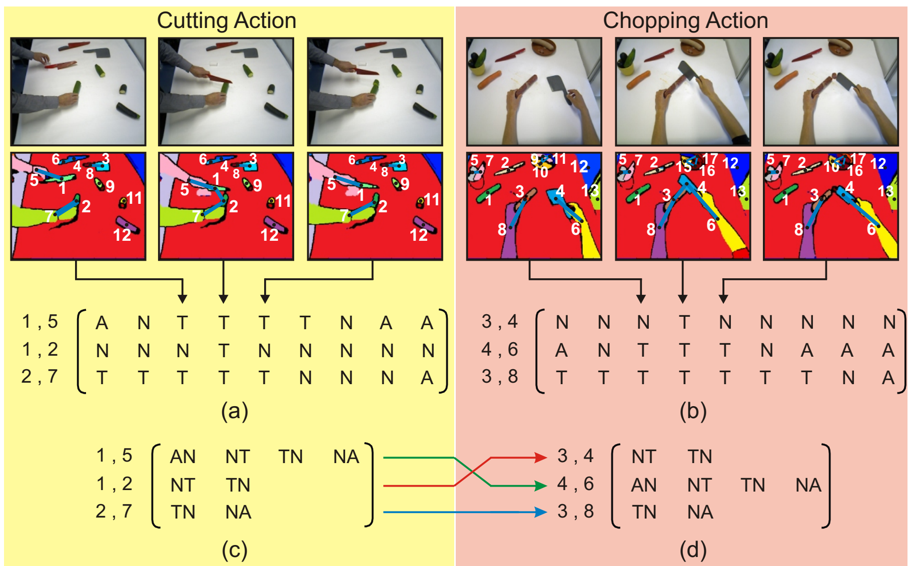


Figure B.14: Two sample manipulation action scenarios: *"Cutting a cucumber with a knife"* (on the left) and *"Chopping a sausage with a cleaver"* (on the right). (a-b) Extracted de-noised SECs (ξ_o) with some sample original key frames including respective segments and main graphs. (c-d) Corresponding compressed SECs (ξ_c). Colored arrows show row matchings.

ξ_c^1 \ ξ_c^2	3 , 4	4 , 6	3 , 8
1 , 5	50%	**100%**	50%
1 , 2	**100%**	50%	0%
2 , 7	0%	50%	**100%**

(a)

ξ_o^1 \ ξ_o^2	1	2	3	4	5	6	7	8	9
1	100	67	67	33	67	33	33	67	67
2	67	100	67	33	67	33	67	33	33
3	67	67	100	67	100	67	33	33	33
4	33	33	67	100	67	33	0	0	0
5	67	67	100	67	100	67	33	33	33
6	67	100	67	33	67	33	67	33	33
7	100	67	67	33	67	33	33	67	67
8	67	33	33	0	33	67	67	100	67
9	67	33	33	0	33	33	33	67	100

(b)

Figure B.15: Similarity matrices between the *Cutting* and *Chopping* samples given in Fig. B.14. (a) Spatial similarity matrix $\zeta_{spatial}$ indicates possible correspondences between rows (see colored arrows in Fig. B.14). (b) Temporal similarity matrix $\zeta_{temporal}$ with LCS matchings indicated in red circles shows correspondences between columns.

$$\xi_c^2 = \begin{bmatrix} r^2_{1,1} & r^2_{1,2} & \cdots & \cdots & \cdots & \cdots & r^2_{1,\gamma_1^2} \\ r^2_{2,1} & r^2_{2,2} & \cdots & \cdots & r^2_{2,\gamma_2^2} & & \\ \vdots & \vdots & \ddots & \vdots & & & \\ r^2_{k,1} & r^2_{k,2} & \cdots & \cdots & \cdots & r^2_{k,\gamma_k^2} & \end{bmatrix},$$

where $r_{i,j}$ represents a relational *change* between a segment pair

$$r_{i,j} \in \{AN, AT, NA, NT, TA, TN\}\ ,$$

where A, N, and T stand for *Not touching*, *Touching*, and *Absence*, respectively. The lengths of the rows are usually different and given by indices γ.

The first step is comparing the rows of the compressed event chains (ξ_c^1 and ξ_c^2) accounting for a possibly shuffling of rows in different versions of the same manipulations. Therefore, each row of ξ_c^1 is compared with each row of ξ_c^2 in order to find the highest similarity. The comparison process searches for equal entries of one row against the other using a standard sub-string search, briefly described next. Assume that we compare the a^{th} row of ξ_c^1 with the b^{th} row of ξ_c^2. If row a is shorter or of equal length than row b ($\gamma_a^1 \leq \gamma_b^2$), the a^{th} row of ξ_c^1 is shifted $\gamma_b^2 - \gamma_a^1 + 1$ times to the right. At each shift its entries are compared with the one of the b^{th} row of ξ_c^2 and we get as a result set $F_{a,b}$ defined as:

$$F_{a,b} = \{f_t :\ t \in [1, \gamma_b^2 - \gamma_a^1 + 1]\}\ ,$$

$$f_t = \frac{100}{\gamma_b^2} \sum_{i=1}^{\gamma_a^1} \delta_i \quad , \tag{B.1}$$

where γ_b^2 is the normalization factor and i is the row index and with

$$\delta_i = \begin{cases} 1 & \text{if } r^1_{a,i} = r^2_{b,i+t-1} \\ 0 & \text{else} \end{cases} \quad , \tag{B.2}$$

where the set $F_{a,b}$ represents all possible similarities for every shift t, given by f_t, which holds the normalized percentage of the similarity calculated between the shifted rows.

As usual for sub-string searches, we are only interested in the maximum similarity of every comparison hence we define:

$$M_{a,b} = \max(F_{a,b}),$$

For the case $\gamma_a^1 > \gamma_b^2$, a symmetrical procedure is performed by interchanging all indices of Eqs. (B.1), (B.2) above.

Spatial similarity values between all rows of ξ_c^1 and ξ_c^2 are stored in a matrix $\zeta_{spatial}$ with size $m \times k$ as

$$\zeta_{spatial} = \begin{bmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,k} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ M_{m,1} & M_{m,2} & \cdots & M_{m,k} \end{bmatrix}.$$

The final similarity value ($\psi_{spatial}$) between the rows of two compressed event chains is calculated by taking the mean value of the highest similarities across both rows and columns of $\zeta_{spatial}$ as

$$\psi_{spatial} = \frac{1}{m} \sum_{i=1}^{m} \max_j (M_{i,j}), \quad j \in [1, \cdots, k], \tag{B.3}$$

if

$$\max_j (M_{i,j}) = \max_t (M_{t,j}), \quad t \in [1, \cdots, m] \quad . \tag{B.4}$$

The spatial similarity matrix $\zeta_{spatial}$ indicates possible correspondences between rows of ξ_c^1 and ξ_c^2 used to compute temporal similarity in the second step. Note that there can be more than one correspondences between each row and all existing permutations need to be considered in the second step, separately. If there is a size differences between event chains, extra rows with no correspondences will be omitted here, but penalty values will then be applied at the end of the second step.

The complete similarity matrix ($\zeta_{spatial}$) between the *Cutting* and *Chopping* samples (ξ_c^1 and ξ_c^2) is given in Fig. B.15 (a) which shows that first row of ξ_c^1, i.e. $1, 5$, corresponds to the second row of ξ_c^2, i.e. $4, 6$. The same reverse relation exists between the second row of ξ_c^1 and the first row of ξ_c^2. Therefore, rows of the second event chain will be resorted by simply interchanging first and second rows to initiate the second step, i.e. temporal analysis of the method.

In the following second step, we use the time sequence, encoded in the order of columns in the original event chains, to find the best matching permutation and thereby arrive at the final semantic similarity. To this end we will now compare columns of resorted ξ_o^2 with that of ξ_o^1. Note that by contrast to rows, columns of event chains are never shuffled unless they represent different types of actions. Therefore, the column orders of type-similar event chains have to be the same. The comparison procedure of columns is very similar to the one for the rows. Since the lengths of the columns are the same, no shift-operator is required and columns are directly compared index-wise. Similarity values between all columns of ξ_o^1 and ξ_o^2 are stored in a matrix $\zeta_{temporal}$ with the size of $u \times v$ as

$$\zeta_{temporal} = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,v} \\ \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,v} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{u,1} & \theta_{u,2} & \cdots & \theta_{u,v} \end{bmatrix} ,$$

where u and v are the lengths of columns in ξ_o^1 and ξ_o^2.

Once similarities between columns are calculated, we use “Longest Common Subsequence, (LCS)” in order to guarantee that the order of columns is the same. LCS is generally used to explore the longest sequence existing in both input samples sequences. Columns of event chains are used as sequences for this task and LCS matching is computed based on similarities in $\zeta_{temporal}$. Since the number of sequences is constant, the problem is solvable in polynomial time by dynamic programming. Fig. B.15 (b) shows $\zeta_{temporal}$ with LCS matchings indicated in red circles for the *Cutting* and *Chopping* samples ξ_o^1 and ξ_o^2.

The temporal similarity value $\psi_{temporal}$ between the columns of two event chains is then calculated by taking the mean value of the similarities given by LCS matching L_i as

$$\psi_{temporal} = \frac{1}{u} \sum_{i=1}^{u} L_i \quad , \tag{B.5}$$

$$L_i = \begin{cases} 100 & \text{if } \theta_{i,j} = 100 \\ 0 & \text{else} \end{cases} \quad , \tag{B.6}$$

where i and j are the matching column indices between ξ_o^1 and ξ_o^2.

Note that due to noisy segmentation and tracking, size of ξ_o^1 and ξ_o^2 can be different. Therefore, size differences between event chains are used as a penalty to prevent false similarities. The final semantic similarity is then computed as

$$\psi_{final} = \frac{r_1 c_1 \psi_{temporal}}{r_1 c_1 + \dfrac{r_2 c_2 - r_1 c_1}{\rho}} \quad , \quad r_1 < r_2 \quad and \quad c_1 < c_2 \quad , \tag{B.7}$$

where ρ is the penalty value and r_1, c_1, r_2, and c_2 are the number of rows and columns of ξ_o^1 and ξ_o^2, respectively. The final ψ_{final} value between the *Cutting* and *Chopping* samples in Fig. B.15 is calculated as 78% by using Eqs. (B.7), (B.6), and (B.7) with $\rho = 1$. The best matching permutation is further used for categorizing objects as described in [7].

Appendix C. Model Updating

Let ξ_m and ξ_n be two matrices representing a SEC model and a new SEC sample with sizes of $p \times q$ and $k \times l$, respectively. The two matrices can be written as

$$\xi_m = \begin{bmatrix} r^m_{1,1} & r^m_{1,2} & \cdots & r^m_{1,q} \\ r^m_{2,1} & r^m_{2,2} & \cdots & r^m_{2,q} \\ \vdots & \vdots & \ddots & \vdots \\ r^m_{p,1} & r^m_{p,2} & \cdots & r^m_{p,q} \end{bmatrix} \quad and \quad \xi_n = \begin{bmatrix} r^n_{1,1} & r^n_{1,2} & \cdots & r^n_{1,l} \\ r^n_{2,1} & r^n_{2,2} & \cdots & r^n_{2,l} \\ \vdots & \vdots & \ddots & \vdots \\ r^n_{k,1} & r^n_{k,2} & \cdots & r^n_{k,l} \end{bmatrix} \quad ,$$

where $r_{i,j} \in \{A, N, T\}$ is representing the spatial relations between each segment pair as described in section 3.3.

Each model ξ_m is initially assigned with a set of weights $\mathcal{W}$ as

$$\mathcal{W} = [w_1, w_2, \cdots, w_p]^T \quad , \tag{C.1}$$

for representing the appearance frequency of each row, which leads to extraction of all common rows observed in most of SEC samples. Each weight value w_i is initialized to 1. We first compare each row of ξ_n with each row of ξ_m to find identical matches and to further increment the corresponding weight values of the matched rows again by 1. This step is required since rows can be shuffled in the new observation sample ξ_n. While comparing rows, we search for only equal relational changes rather than temporal lengths of relations as explained in Appendix B. In the case of $k > p$, all novel rows observed in ξ_n will then be appended to ξ_m with weight values $\{w_{p+1}, \cdots, w_k\}$ initialized to 1. Common rows are then those with weights higher than $\frac{|\mathcal{W}|}{2}$. Next, the order of rows in ξ_n is resorted considering the order of their best matches with common rows in ξ_m. The sorting process yields the same row numbers in ξ_n and ξ_m, which is required for analyzing columns as described next.

The following step covers the temporal information embedded in the columns of ξ_n and ξ_m, and is similar to the previous approach explained for rows. We here assume that each column in an event chain is a *state* defining one action primitive. Hence, we seek for all primitives derived from new observations and compute the transition between them. Let $\mathcal{S}_m$ be a set of existing states in the current model ξ_m :

$$\mathcal{S}_m = \{s_1, s_2, \cdots, s_q\} \quad , \tag{C.2}$$

where each $s_i = \{r^m_{j,i} : \ j \in [1, \cdots, p]\}$. We now compare each state, i.e. column, in the sorted version of ξ_n with those in ξ_m by employing the same approach as defined for the temporal analysis in Appendix B. In the case of having more states in ξ_n, i.e. $l > q$, all novel states are appended to $\mathcal{S}_m$, and then transitions between each state are calculated. We assign a probability value $P_{i,j}$ defining the transition from s_i to s_j, which is incremented when two states are consecutive, i.e. $s_j = s_{i+1}$ in ξ_n.

Following state transition calculation, the learned model ξ_m is refined with the new states $\hat{\mathcal{S}}_m$ having the maximum transitions between each; that is,

$$\hat{\mathcal{S}}_m = \{s_{\alpha_1}, s_{\alpha_2}, \cdots, s_{\alpha_l}\} \quad , \tag{C.3}$$

$$\alpha_{t+1} = \arg\max_j \ (P_{\alpha_t, j}) \quad , \tag{C.4}$$

where $\alpha_0 = 1$ is for the initial state and $P_{i,j} = 0$ is the termination condition of the state sequence.

Note that in the process of creating a new model, $\hat{\mathcal{S}}_m$ will directly be equal to the states of ξ_n. In the case of merging similar models, i.e. those with high semantic similarity, one of the models will be assumed as ξ_n to employ the same refinement procedure explained above. It is also important to note that all SEC samples used for updating the same model ξ_m will be assigned with the same cluster label which yields self-clustering of observed SEC samples.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience.

References

[1] S. Schaal, Is imitation learning the route to humanoid robots?, Trends in Cognitive Sciences 3 (1999) 233–242.

[2] A. Billard, S. Calinon, F. Guenter, Discriminative and adaptive imitation in uni-manual and bi-manual tasks, Robot. Auton. Syst. 54 (2006) 370–384.

[3] M. Pardowitz, S. Knoop, R. Dillmann, R. D. Zöllner, Incremental Learning of Tasks From User Demonstrations, Past Experiences, and Vocal Comments, IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics 37 (2) (2007) 322–332.

[4] S. Ekvall, D. Kragic, Robot learning from demonstration: a task-level planning approach, International Journal of Advanced Robotic Systems 5 (3) (2008) 223–234.

[5] R. Cubek, W. Ertel, Learning and Execution of High-Level Concepts with Conceptual Spaces and PDDL, in: 3rd Workshop on Learning and Planning, ICAPS (21st International Conference on Automated Planning and Scheduling), 2011.

[6] E. E. Aksoy, A. Abramov, F. Wörgötter, B. Dellen, Categorizing object-action relations from semantic scene graphs, in: IEEE International Conference on Robotics and Automation (ICRA), 2010, pp. 398–405.

[7] E. E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, F. Wörgötter, Learning the semantics of object-action relations by observation, The International Journal of Robotics Research 30 (10) (2011) 1229–1249.

[8] E. E. Aksoy, B. Dellen, M. Tamosiunaite, F. Wörgötter, Execution of a dual-object (pushing) action with semantic event chains, in: Proceedings of 11th IEEE-RAS International Conference on Humanoid Robots, 2011, pp. 576–583.

[9] 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 International Conference on Development and Learning and Epigenetic Robotics (ICDL-EPIROB), 2013.

[10] S. Niekum, S. Chitta, A. Barto, B. Marthi, S. Osentoski, Incremental semantically grounded learning from demonstration, Robotics Science and Systems (RSS) 2013.

[11] N. Badler, Temporal scene analysis: Conceptual descriptions of object movements, Ph.D. thesis, University of Toronto, Canada (1975).

[12] 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.

[13] M. Sridhar, G. A. Cohn, D. Hogg, Learning functional object-categories from a relational spatio-temporal representation, in: Proc. 18th European Conference on Artificial Intelligence, 2008, pp. 606–610.

[14] H. Kjellström, J. Romero, D. Kragić, Visual object-action recognition: Inferring object affordances from human demonstration, Comput. Vis. Image Underst. 115 (1) (2011) 81–90.

[15] Y. Yang, C. Fermüller, Y. Aloimonos, Detection of manipulation action consequences (mac), in: International Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2563–2570.

[16] D. Summers-Stay, C. Teo, Y. Yang, C. Fermuller, Y. Aloimonos, Using a minimal action grammar for activity understanding in the real world, in: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, 2012, pp. 4104–4111.

[17] H. S. Koppula, R. Gupta, A. Saxena, Learning human activities and object affordances from rgb-d videos, The International Journal of Robotics Research 32 (8) (2013) 951–970.

[18] D. Martinez, G. Alenya, P. Jimenez, C. Torras, J. Rossmann, N. Wantia, E. E. Aksoy, S. Haller, J. Piater, Active learning of manipulation sequences, (in press), in: IEEE International Conference on Robotics and Automation (ICRA), 2014.

[19] G. Luo, N. Bergstrom, C. Ek, D. Kragic, Representing actions with kernels, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011, pp. 2028–2035.

[20] R. Vuga, E. E. Aksoy, F. Wörgötter, A. Ude, Augmenting semantic event chains with trajectory information for learning and recognition of manipulation tasks, in: 22nd International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD), 2013.

[21] M. Wächter, S. Schulz, T. Asfour, E. E. Aksoy, F. Wörgötter, R. Dillmann, Action sequence reproduction based on automatic segmentation and object-action complexes, in: IEEE/RAS International Conference on Humanoid Robots (Humanoids), 2013.

[22] J. Papon, T. Kulvicius, E. E. Aksoy, F. Wörgötter, Point cloud video object segmentation using a persistent supervoxel world-model, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013.

[23] M. J. Aein, E. E. Aksoy, M. Tamosiunaite, J. Papon, A. Ude, F. Wörgötter, Toward a library of manipulation actions based on semantic object-action relations, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013.

[24] K. Pauwels, N. Krüger, M. Lappe, F. Wörgötter, M. M. Van Hulle, A cortical architecture on parallel hardware for motion processing in real time, Journal of Vision 10.

[25] J. Papon, A. Abramov, E. E. Aksoy, F. Wörgötter, A modular system architecture for online parallel vision pipelines, in: IEEE Workshop on Applications of Computer Vision (WACV), 2012, pp. 361–368.

[26] A. Abramov, K. Pauwels, J. Papon, F. Wörgötter, B. Dellen, Depth-supported real-time video segmentation with the kinect, in: Applications of Computer Vision (WACV), 2012 IEEE Workshop on, 2012, pp. 457–464.

[27] A. Abramov, E. E. Aksoy, J. Dörr, K. Pauwels, F. Wörgötter, B. Dellen, 3d semantic representation of actions from efficient stereo-image-sequence segmentation on GPUs, in: 5th International Symposium 3D Data Processing, Visualization and Transmission, 2010, pp. 1–8.

[28] N. Otsu, A Threshold Selection Method from Gray-level Histograms, IEEE Transactions on Systems, Man and Cybernetics 9 (1) (1979) 62–66.

[29] J. Piaget, The Origins of Intelligence in the Child, Routledge, London, New York, 1953.

[30] E. E. Aksoy, M. Tamosiunaite, F. Wörgötter, Decomposition of long manipulation actions (under review), Computer Vision and Image Understanding.

[31] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local svm approach, in: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, Vol. 3, 2004, pp. 32–36 Vol.3.

[32] A. Gupta, L. Davis, Objects in action: An approach for combining action understanding and object perception, in: Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, 2007, pp. 1–8.

[33] F. Wörgötter, E. E. Aksoy, N. Krüger, 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.