Structural bootstrapping at the sensorimotor level for the fast acquisition of action knowledge for cognitive robots

E. E. Aksoy[1], M. Tamosiunaite[1], R. Vuga[2], A. Ude[2], C. Geib[3], M. Steedman[3], and F. Wörgötter[1]

***Abstract*—Autonomous robots are faced with the problem of encoding complex actions (e.g. complete manipulations) in a generic and generalizable way. Recently we had introduced the Semantic Event Chains (SECs) as a new representation which can be directly computed from a stream of 3D images and is based on changes in the relationships between objects involved in a manipulation. Here we show that the SEC framework can be extended (called "extended SEC") with action-related information and used to achieve and encode two important cognitive properties relevant for advanced autonomous robots: The extended SEC enables us to determine whether an action representation (1) needs to be newly created and stored in its entirety in the robot's memory or (2) whether one of the already known and memorized action representations just needs to be refined. In human cognition these two processes (1 and 2) are known as accommodation and assimilation. Thus, here we show that the extended SEC representation can be used to realize these processes originally defined by Piaget for the first time in a robotic application. This is of fundamental importance for any cognitive agent as it allows categorizing observed actions in *new* versus *known* ones, storing only the relevant aspects.**

I. Introduction

A central issue for the development of autonomous robots is how to quickly acquire new concepts for planning and acting, 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 methods. Especially the Learning from Demonstration (LfD) paradigm seems promising for cognitive learning ([1], [2], [3], [4], [5]) because we (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 to infer the "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], [8]. The SEC framework analyzes the sequence of changes of the *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].

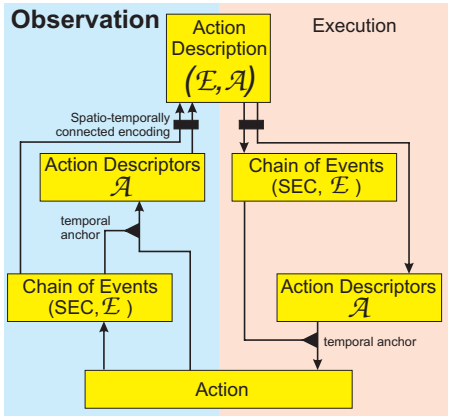

Fig. 1: Schematic representation of the acquisition of action information by observation using SECs and Action descriptors (left) as well as execution (right).

The goal of this paper is to extend the SECs with action related information (action descriptors) and to use the obtained structure for the assimilation of novel information into the existing schemata or for the creation of novel schemata (accommodation) in a Piagetian sense [9]. The first happens when an agent finds that a newly observed action is compatible with an already memorized SEC, but there are some elements present in the new action which are truly novel. These can then be stored (assimilated) together with the known ones into the existing schema. The second happens when the agent realizes that the new action does not compare to any of its known schemata and requires a novel schema to be created (accommodation). This way agent's cumulative memory of actions can be developed. The main contribution of this paper is therefore the enrichment process of event chains to further use memory in a more efficient way. With respect to our previous approaches ([6], [7], [8]), enriched SECs also lead to extraction and comparison of action descriptors such as trajectory segments, pose, and object information.

The whole process of action representation using SECs and action descriptors is summarized in Fig. 1. In this paper we are concerned only with the observation phase (left side). The

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

[1]Inst. Physics-3 & BCCN, University of Göttingen, Friedrich-Hund Platz 1, D-37077, Germany [eaksoye,minija,worgott] at physik3.gwdg.de

[2]Jožef Stefan Institute, Department of Automatics, Biocybernetics and Robotics, Jamova 39, Ljubljana, Slovenia [rok.vuga,ales.ude] at ijs.si

[3]School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh, Scotland [cgeib,steedman] at inf.ed.ac.uk

Fig. 2: Real action scenarios. (a), (c), (e) Sample original key frames, (b), (d), (f) corresponding segments and graphs for the following actions: *Cutting*, *Chopping*, and *Stirring*.

execution stage (right side) is described in [10] which shows the possibility of imitating actions with robots by directly using the here introduced representations.

The paper is organized as follows. We start with the description of the extended SEC representation and then provide an example for assimilation as well as accommodation using this framework. We call the developed algorithm Structural Bootstrapping. In the discussion section we embed these results into the state of the art and provide also a comparison to child language development, from where the concept of Bootstrapping originates.

II. Data and Data Representations

A. Data and Pre-processing

Data structures and algorithms we developed are generic and do not depend on the actual input data. Nonetheless, it is best to first describe some example experiments, which should make it is easier to understand all components of our representation. We have investigated three different manipulation actions: *Cutting*, *Chopping*, and *Stirring*. In the *Cutting* action, a hand is cutting a cucumber by moving a knife back and forth. In the *Chopping* action, a cleaver follows a straight trajectory to cut a carrot. The *Stirring* action represents a scenario in which a spoon is used to stir milk in a bucket. We recorded these three manipulation sequences with Microsoft Kinect. The developed system first pre-processes all movie frames by a real-time image segmentation procedure ([11], [12]) to uniquely identify and track objects (including hands) in the observed actions. Each segmented image is represented by a graph: nodes represent segment centers and edges indicate whether two objects touch each other or not (in 3D). Fig. 2 (a-f) depict sample original images with extracted segments (regions) and graphs for each scenario.

While recording each action sequence, the trajectories of the hand and manipulated objects as well as their poses are measured by the 3D motion capture system Optotrak. Fig. 5 illustrates the measured trajectories for the knife and cleaver as used in the *Cutting* and *Chopping* scenario. These trajectories were measured by attaching a set of 3 active markers to each of the objects involved in the action.

1) Semantic Event Chain $\mathcal{E}$: 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 $\mathcal{E}$, which is a matrix where rows (index i) are possible pairwise object relations (e. g. between the hand and knife or the knife and cucumber) and columns (index j) describe the scene configuration at time j when a new main graph has occurred. Fig. 3 (a) indicates the SEC with sample main graphs derived from the *Cutting* action shown in Fig. 2 (a).

Let $\mathcal{E}$ be a semantic event chain with size $n \times m$. Then it can be written as:

$$\mathcal{E} = \begin{bmatrix} R(o_{a_1}, o_{b_1}) \\ R(o_{a_2}, o_{b_2}) \\ \vdots \\ R(o_{a_n}, o_{b_n}) \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,m} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n,1} & r_{n,2} & \cdots & r_{n,m} \end{bmatrix}, \quad (1)$$

where $r_{i,j}$ represents a spatial relation R between an object pair o_{a_i}, o_{b_i} at time j. Thus all pairs of objects need to be considered once, where rows that do not contain any changes in object-object relations are deleted. The maximum total number of rows n is defined as $n = \lambda(\lambda - 1)/2$, where λ is the total number of segments. However, the total number of columns m depends on the action and can vary.

Relations are given by:

$$r_{i,j} \in \{\text{not touching (N), touching (T), absence (A)}\}, \quad (2)$$

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 absence of an object yields A.

2) Action Encoder Matrix $\mathcal{A}$: A central advantage of our framework is that we can extract temporal anchor points from a SEC. These points tell us when to "pay attention to the action", because action-relevant details occur at or near the *transitions* between the relations recorded in the SEC. These transitions are encoded by $\mathcal{T}_{i,j}$, defined as:

$$\mathcal{T}_{i,j} = \begin{cases} 0 & \text{if } r_{i,j} = r_{i,j-1},\ j > 1 \\ [\mathcal{T}.\{d^1, d^2, \ldots, d^k\}]_{i,j} & \text{else} \end{cases} \quad (3)$$

The variables $\mathcal{T}_{i,j}$ correspond to the respective transition and its k action descriptors d (described later). One could think of each $\mathcal{T}$ as a derivative-like "change-encoder", which is non-zero whenever there is a change in the scene graph ("something has happened with any of the objects"). For improved readability transitions are given in plain-text (e.g. NT, AT, AN, etc.) using the corresponding pairs of relations r from the event chain to encode this. For example, entries $(r_{4,4}, r_{4,5})$ and $(r_{4,7}, r_{4,8})$ represent transitions from N to T and from T to N as depicted in shaded boxes in Fig. 3 (a). Hence in these cases we would write specifically $\mathcal{T}_{4,5} = [\text{NT}]_{4,5}$ and $\mathcal{T}_{4,8} = [\text{TN}]_{4,8}$. For these (and all others where $r_{i,j} \neq r_{i,j-1}$) we have $\mathcal{T}_{i,j} \neq 0$. Descriptors d need to be computed next.

Note that for the first column ($j = 1$) there are no transitions and we write $\mathcal{T}_{i,1} = \text{X}_i, \forall i$, where descriptors $d_{i,1}$ define the initial state of the corresponding objects before the action progresses. Fig. 3 (b) depicts the corresponding transitions derived from the SEC given in Fig. 3 (a). In the following we will abbreviate $\mathcal{T}.\{d^1, d^2, \ldots, d^k\}$ with $\mathcal{T}.d$ where possible.

The resulting structure will, thus, be a matrix describing the action $\mathcal{A}$:

$$\mathcal{A} = \begin{bmatrix} \text{X}_1 & [\mathcal{T}.d]_{1,2} & \cdots & [\mathcal{T}.d]_{1,m} \\ \text{X}_2 & [\mathcal{T}.d]_{2,2} & \cdots & [\mathcal{T}.d]_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \text{X}_n & [\mathcal{T}.d]_{n,2} & \cdots & [\mathcal{T}.d]_{n,m} \end{bmatrix}. \quad (4)$$

Each descriptor d contains information about the objects involved, their relative poses at time j of the event, as well as their trajectories until time j and the forces involved. In our implementation the variable k from Eq. (3) is set to 4, but this can be changed if more action-relevant attributes are needed. We define the descriptor set as:

- $d^1_{i,j} = \{o_{a_i}, o_{b_i}\}_i$ is a set containing two object identifiers of those objects that are involved in the given event. Note that by definition of the event chain there are always exactly two objects for each event. Objects do not change along the rows of an event chain, thus index j is irrelevant.
- $d^2_{i,j} = \{p\}_{i,j} = \{x, y, z, \alpha, \beta, \gamma\}_{i,j}$ is a set containing *relative* pose information *between* two object identifiers. The x, y, z and α, β, γ values hold corresponding translation and rotation values, respectively.
- $d^3_{i,j} = \{t\}_{i,j} = \{\mathbf{s}, \mathbf{g}, \tau, \mathbf{w}_{1,\ldots,6}\}_{i,j}$, $\mathbf{s}$, $\mathbf{g} \in \mathbb{R}^6$, is

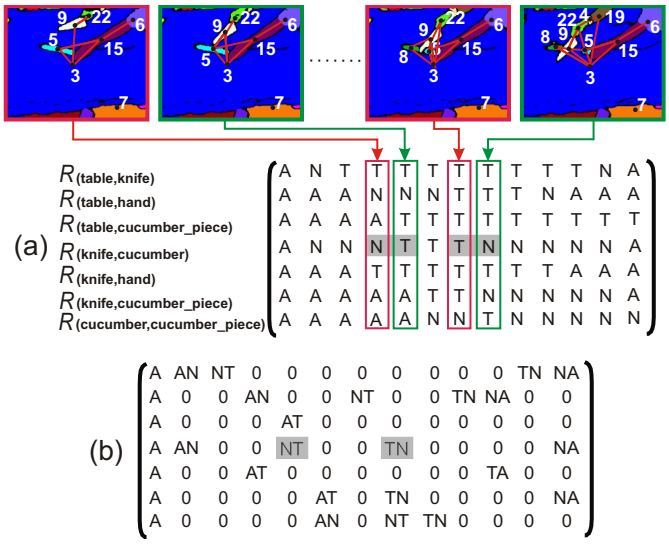

Fig. 3: The SEC ($\mathcal{E}$) and transition matrix extracted from the *Cutting* action given in Fig. 2 (a). (a) SEC with corresponding sample main graphs and segments. (b) Transition matrix showing the respective relational transitions between object pairs. First column defines the initial relations. Shaded boxes show two sample transitions: $\mathcal{T}_{4,5} = [\text{NT}]_{4,5}$ and $\mathcal{T}_{4,8} = [\text{TN}]_{4,8}$.

a set of parameters containing trajectory information in the Cartesian task space. In this study we use the modified Dynamic Movement Primitives (DMPs, [13]) to encode movement trajectories because they have faster convergence at the end points compared to the standard DMPs and allow smooth joining of movement sequences. Variables $\mathbf{s}$ and $\mathbf{g}$ denote start and goal (end) points of the DMP, respectively, and τ is the time constant modulating the speed of movement. Vectors $\mathbf{w}_l$, $l = 1, \ldots, 6$, hold the shape parameters of the DMP given as weights for about 5-20 Gaussian kernels (see [13] for a description of the DMP parameterization). DMPs offer the advantage that they are robust to perturbations, can generalize to different start- and end-points, and also allow online modification of the movement by ways of sensory coupling ([14], [15], [13], [16]).

- $d^4_{i,j} = \{\mathbf{f}\}_{i,j} = \{f_x, f_y, f_z, \tau_x, \tau_y, \tau_z\}_{i,j}$ is a 6D vector containing the Cartesian space force and torque information. Force information cannot directly be obtained from human demonstration and but requires own exploration (similar to the situation for a human child).

Thus, derived from the event chain $\mathcal{E}$ and using additional information encoded with descriptors d, we have now obtained a new matrix $\mathcal{A}$. The event chain becomes obsolete by this. Still, it makes sense to keep both, $\mathcal{E}$ and $\mathcal{A}$, to make the next step, the description of the bootstrapping algorithm, easier.

B. Algorithm: Structural Bootstrapping

Sensorimotor structural bootstrapping consists of four main steps: (1) Initial memory formation, (2) observation, (3) comparison, and (4) generalization via Assimilation or Accommodation (Fig. 4). In the very first step, the framework analyzes and stores an action in the specific format described above. Let this first observed and stored action be $\langle(\mathcal{E}, \mathcal{A})^{a1}\rangle$ where in the pseudo-code below we denote memory storage by brackets $\langle\ \ \rangle$. In the second step, a new action is observed $(\mathcal{E}, \mathcal{A})^{a2}$. In the third step, comparison, we determine whether both, stored and newly observed, actions are semantically the same (for example cutting and chopping have the same SEC, whereas stirring has a different one). Similarity between two event chains, $\mathcal{E}^{a1}$ and $\mathcal{E}^{a2}$ is measured using the definition of spatiotemporal similarity $\zeta(\mathcal{E}^{a1}, \mathcal{E}^{a2})$ for SECs given in [6]. The last step, generalization, is divided into two aspects. If the semantic similarity $\zeta(\mathcal{E}^{a1}, \mathcal{E}^{a2})$ is below a certain threshold $\tau_{\mathcal{E}}$, actions are not type-similar and we will store the complete newly observed action as a new schema (Accommodation, lower path in Fig. 4). Otherwise, i. e. for type-similar actions, we perform a comparison of the two descriptor sets $[\mathcal{T}.d]^{a1}_{i,j}$ and $[\mathcal{T}.d]^{a2}_{i,j}$. Below we describe how DMP descriptors $d^3_{i,j}$ can be compared. The comparison of other descriptors (object identity, relative pose information, and torques at the contact point) can be done using standard metrics.

The acquisition of new action information is completed by extending the memory either by storing the new action or by additionally storing those descriptors that are different compared to those already stored in the memory of a known action.

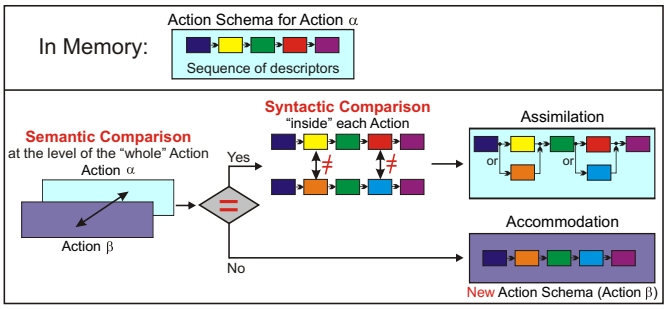

Fig. 4: Schematic representation of the required steps for Structural bootstrapping. For explanation see text.

In the latter case, individual entries in matrix $\mathcal{A}$ turn into tuples of descriptor sets $[\mathcal{T}.(d^{a1}, d^{a2})]_{i,j}$. This whole procedure can be repeated as soon as more actions are observed. A concise description of the entire structural bootstrapping framework is given in Algorithm 1.

Algorithm 1 Sensorimotor Structural Bootstrapping

```
Store first action in memory ⟨···⟩.
⟨ ⟩ = ⟨ ⟩ + (ℰ, 𝒜)^a1 with 𝒜^a1 defined by [𝒯.d]^a1_{i,j}
Observe next action.
(ℰ, 𝒜)^a2 with 𝒜^a2 defined by [𝒯.d]^a2_{i,j}
Semantic Comparison.
ζ_ℰ = ζ(ℰ^a1, ℰ^a2)
if ζ_ℰ < τ_ℰ (small similarity!) then
  New Action! Create new memory (Accommodate).
  ⟨(ℰ, 𝒜)^a1⟩ = ⟨(ℰ, 𝒜)^a1⟩ + (ℰ, 𝒜)^a2
else
  Type-similar Action! Perform syntactic comparison.
  for each event 𝒯_{i,j} ≠ 0 do
    for all descriptors d indexed by l do
      ζ_{d^l} = ζ([𝒯.d^l]^a1_{i,j}, [𝒯.d^l]^a2_{i,j})
      Compare syntactic similarity.
      if ζ_{d^l} < τ_{d^l} (small similarity!) then
        New element! Assimilate into existing memory.
        ⟨(ℰ, 𝒜)^a1⟩ with [𝒯.d^l]^a1_{i,j} = [𝒯.(d^{l,a1}, d^{l,a2})]^a1_{i,j}
      end if
    end for
  end for
end if
```

Algorithm 1 requires that we can compare actions both at the semantic event chain level [6] and at the action descriptor level. In the current implementation we use object identities, relative poses, and trajectories for comparisons at the action descriptor level. While comparing identities and relative poses is rather straightforward, the comparison of trajectories is more difficult and is described in more detail below.

1) Comparison of movements encoded by DMPs: DMPs provide a temporary and spatially invariant representation of a movement. Even if the timing τ and the absolute position of the movement ($\mathbf{s}$ and $\mathbf{g}$) in space change, the parameters $\mathbf{w}_l$

stay the same. Thus trajectories with similar velocity profiles will be fitted by similar shape parameters $\mathbf{w} = [\mathbf{w}_1^T, \ldots, \mathbf{w}_6^T]^T$ [14]. Similarities between two trajectories encoded by DMPs can be measured by computing the correlation between their parameter vectors. The correlation is given by the cosine of the angle between these two vectors:

$$\frac{\mathbf{w}_1^T \mathbf{w}_2}{\|\mathbf{w}_1\| \|\mathbf{w}_2\|}, \tag{5}$$

where $\mathbf{w}_1$ and $\mathbf{w}_2$ are the parameter vectors of two different movements. Thus, in the training phase we store a set of prototype movements for each action primitive. Classification is then performed by comparing the newly observed trajectory extracted from SECs to the available prototypes. More advanced methods like support vector machines could be used, but this was not necessary in our experiments.

The proposed classification method can be applied to compare DMPs describing the movements as long as the shape of underlying motion trajectories does not change with respect to the current configuration of the task. If this is not the case, then we can use statistical generalization with respect to the parameters of the task [17] to generate new movement prototypes to which the newly observed trajectories can be compared.

III. Results

We have applied the structural bootstrapping algorithm described above to the three example actions (*Cutting, Chopping, Stirring*). The framework first extracts key events and generates SECs for all action sequences as explained in section II-A. Then action-encoder matrices ($\mathcal{A}$) are determined. Fig. 5 shows an incomplete, while graphical, rendering of the action-encoder matrices for the *Cutting* and *Chopping* actions.

We use *Cutting* as our reference action (Action *cut*, see also "In Memory", Fig. 4) and commit it to memory $\langle\ \rangle = \langle\ \rangle + (\mathcal{E}, \mathcal{A})^{cut}$. Then we define *Chopping* and *Stirring* as action indices *chop* and *stir*, respectively.

Structural bootstrapping continues with a semantic comparison of the event chains in the spatiotemporal domain. Fig. 6 (a) illustrates the similarity values for the different actions. Similarity measures are basically computed by comparing rows and columns of two event chains using simple sub-string search and counting algorithms. Relational changes are considered while comparing the rows, whereas for the columns the temporal order counts. We first search for the correspondences between rows of two event chains since rows can be shuffled. The searching process compares and counts equal entries of one row against the other using a standard sub-string search which does not rely on dimensions and allows to compare arbitrarily long manipulation actions. We then examine the order of columns to get the final similarity result. Details for similarity calculations are given in [6].

If one compares *Cutting* with itself, similarity is of course 100%, but we also observe high similarity values (88%) between *Cutting* and *Chopping*. On the other hand, the similarity between *Cutting* and *Stirring* is only (55%). In our earlier studies we had measured the discriminability of our

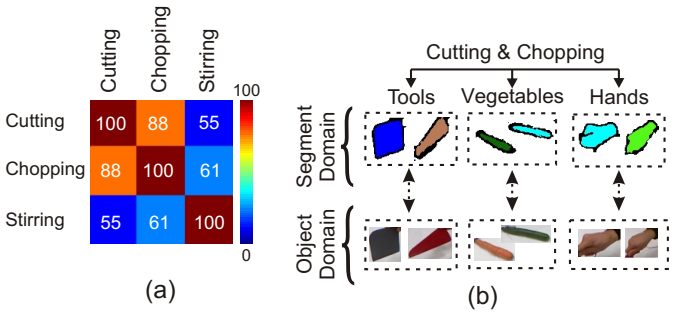

Fig. 6: Semantic comparison. (a) Similarity values between the *Cutting*, *Chopping*, and *Stirring* actions. (b) Segment categories showing which segments exhibit the same role in type-similar actions.

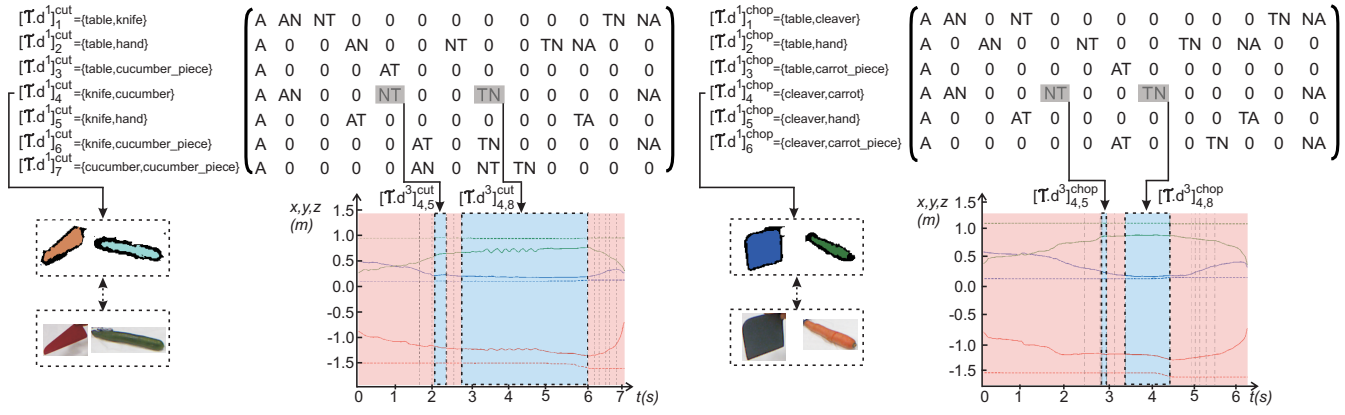

Fig. 5: Action-encoder matrices ($\mathcal{A}$) with extracted descriptors for the *Cutting* and *Chopping* actions. Movement is described in table coordinate system, x and y - table plane coordinates (red and green), z - distance from the table (blue); solid lines stand for the tool demonstrator's hand holding, dashed lines for the tip of the cucumber.

similarity measure using identical actions performed somewhat differently in a noisy environment [6]. From these studies we know that the discriminative threshold is usually about $\tau_{\mathcal{E}} = 65\%$. When applying this threshold here we find that *Cutting* and *Chopping* are regarded as similar, whereas *Cutting* and *Stirring* are not. First we observe that this matches to our lay-man's expectations. In general, we observe that this type of classification renders human-compatible semantics of "same/similar" versus "different" actions. [18] provides a huge confusion matrix showing the semantic similarities between different actions (e.g. push, hide, uncover, stir, cut, etc.) based on the manipulation action ontology. This huge confusion matrix shows the here presented semantic representation can distinguish actions to initiate the bootstrapping process.

The bootstrapping algorithm then proceeds differently for different actions. For *Stirring* we perform Accommodation and just commit the complete descriptor set to memory, i. e. $\langle(\mathcal{E},\mathcal{A})^{cut}\rangle = \langle(\mathcal{E},\mathcal{A})^{cut}\rangle + (\mathcal{E},\mathcal{A})^{stir}$.

For the type-similar actions *Cutting* and *Chopping* we perform a (syntactic) comparison at the level of the individual descriptors d. First we consider d^1, the objects. We find several objects (see Fig. 6 (b)). Note that noisy segment groups observed in some action versions are not categorized as objects since they can all be ignored after applying a SIFT-based object recognition algorithm [19] in the segment domain. Also different trajectories are observed. Fig. 5 shows sample object and trajectory descriptors computed for the *Cutting* and *Chopping* scenarios. For instance, as indicated by the shaded boxes, in both actions the same relational transitions (i.e. events) are observed from N to T and from T to N at index numbers $i = 4, j = 5$ and $i = 4, j = 8$. Here we can perform Assimilation. Specifically $[\mathcal{T}.d^1]^{cut}_{4,j} = \{knife, cucumber\}$ and $[\mathcal{T}.d^1]^{chop}_{4,j} = \{cleaver, carrot\}$. Assimilation renders: $\langle(\mathcal{E},\mathcal{A})^{cut}\rangle$ with $[\mathcal{T}.d^1]^{cut}_{4,j} = [\mathcal{T}.(d^{1,cut}, d^{1,chop})]^{cut}_{4,j}$. Here we note that the "concept of cutting" $[\mathcal{T}.d^{cut}]^{cut}$ is extended by "aspects of chopping" $[\mathcal{T}.(d^{cut}, d^{chop})]^{cut}$ (denoted by the general versus the specific indices in this notation).

Furthermore, Fig. 7 shows that relative poses of manipulated objects are rather weak cues to distinguish the cutting and chopping actions. The main reason is that the pose of a cutting tool is usually constrained to be perpendicular and in the middle of the object to be cut. In this sense, no assimilation is needed for pose. Note that each sample in Fig. 7 is recorded when the cutting tool starts to touch the object to be cut. However, the measured trajectory descriptors $[\mathcal{T}.d^3]_{4,8}$, as indicated with blue boxes in Fig. 5, are highly different in both actions as a consequence of the nature of cutting and chopping actions. Fig. 8 shows a correlation matrix between different instances of these two trajectory samples according to equation (5). On the other hand, descriptors $[\mathcal{T}.d^3]_{4,5}$ in both actions are quite similar since in both versions the hand is approaching to vegetables in a similar way. Therefore, only $[\mathcal{T}.d^3]^{chop}_{4,8}$ is added to the memory yielding: $\langle(\mathcal{E},\mathcal{A})^{cut}\rangle$ with $[\mathcal{T}.d^3]^{cut}_{4,8} = [\mathcal{T}.(d^{3,cut}, d^{3,chop})]^{cut}_{4,8}$.

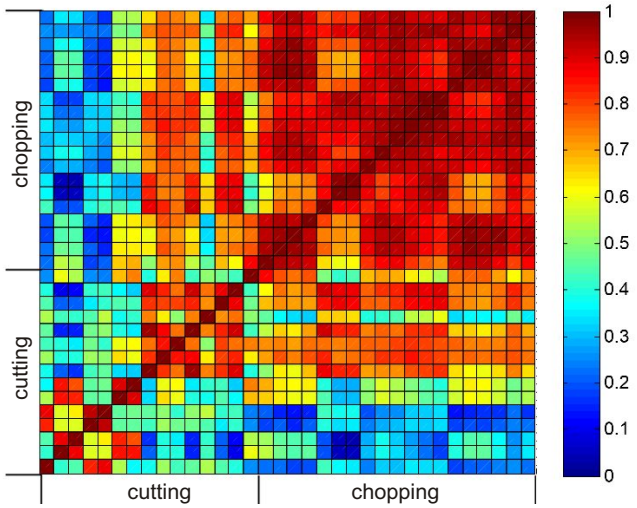

Fig. 7: Correlation between relative poses of the manipulated objects in 15 instances of cutting and 20 instances of chopping actions. Each instance is recorded when the cutting tool starts to touch the object to be cut. Red corresponds to the maximum correlation of 1.0 between the sample pair of poses and blue corresponds to the correlation of 0.0.

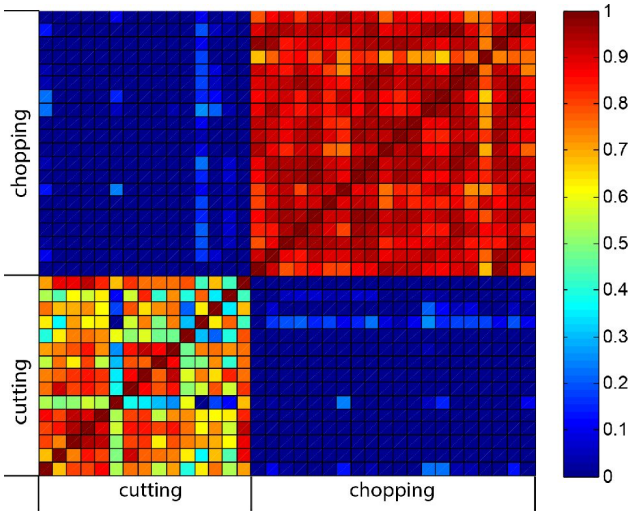

Fig. 8: Correlation between 15 instances of cutting and 20 instances of chopping trajectories according to equation (5). In red and blue are indicated the maximum (1.0) and minimum (0.0 or below) correlations between two sample trajectories.

IV. Discussion and Conclusion

In this paper we have presented two complementary approaches. (1) We have extended the Semantic Event Chain framework by action descriptors and (2) we have used the new framework to compare actions at different levels of semantic depth. This allowed us to subsume cutting and chopping into the same action category (still named "cut") allowing to share most of the cutting- and chopping-action description within the same memory structure. On the other hand we were also

able to realize that cutting and stirring are more fundamentally different such that for both different memory representations have to be stored in their entirety. This distinction arises from a structural comparison either at the level of the SECs or "inside" the action descriptors d, which is called here Structural Bootstrapping. So far this study is based on only three actions. This is due to the fact that all these experiments take quite long and require storing various action-information combining different methods. Our current goal was to show the principles of accommodation and assimilation for which three actions suffice, but we are currently in the process of developing a more complete action-library based on the here presented encoding principles. Next we will now try to embed our study in the state of the art to show similarities and differences to other approaches.

A. State of the Art - Action Classification

Learning from Demonstration (LfD) has been successfully applied both at the control [1], [2] as well as the symbolic level [3], [4], [5]. Although various types of actions can be encoded at the control level, i. e. 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 the resulting representation. Although our approach with SECs is a symbolic-level representation, SECs are 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, [20] introduced the first approach about the 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 continuous perception of actions is ignored and is substituted instead by the idealized hand-made image sequences. This, however, had not been pursued in the field any longer as only now powerful enough image processing methods became available.

Thus, still there are only a few approaches attempting to reach the semantics of manipulation actions in conjunction with the manipulated objects [21], [22], [23]. The work presented in [21] represents an entire manipulation sequence by an activity graph which holds spatiotemporal object interactions. The difficulty is, however, that complex and large activity graphs need to be decomposed for further processing. In the work of [22], 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 is then used to compute the correlation between objects and manipulations. [23] introduced visual semantic graph (inspired from our scene graphs) to recognize action consequences based on changes in the topological structure of the manipulated object. 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.

[24] 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. In our case, each relational transition is considered as a temporal anchor point, at which additional action descriptors are stored. All temporal anchor points are then used in the bootstrapping algorithm. In this sense, to our best knowledge, our work is the first attempts to evaluate semantics of manipulations in a Piagetian sense.

B. State of the Art - Link to Child Language Development

Apart from the aspect of action classification, there is another important link existing of our work to – in this case – an entirely different field, much unrelated to robotics. Structural Bootstrapping, as shown here, is strongly influenced from processes that dominate child language acquisition.

Children acquire the meaning of new words and constructions in their language using two related mechanisms. The primary process is *semantic bootstrapping* where the child associates "meaning-from-the-outside-world" with components of sentences. For example, if the word open is consistently uttered in situations where opening occurs (whatever else is going on), then the meaning of the word can be probabilistically inferred from the conceptual representation of the observed event ([25]). Once a certain amount of language has been acquired, a second process of syntactic bootstrapping can speed up this process by exploiting structural similarity between linguistic elements. This can take place even entirely within language (hence in a purely symbolic way without influence from the world). The most probably meaning of a new word can be estimated on the basis of the prior probability established by previously encountered words of the same semantic and syntactic type in similar syntactic and semantic contexts. For example, if a child knows the meaning of "open the box" and then hears the sentence "open the closet", it can infer that a "closet" denotes a thing that can be opened (rather than a word meaning the same thing as "open") without ever having seen one ([25], [26] see [27] for a comparison between semantic and syntactic bootstrapping). Essentially this amounts to inference of the syntactic and semantic type of an unknown word from its grammatical role and the surrounding context of probabilistically known words. These two generalization mechanisms are very powerful and allow young humans to acquire language without explicit instruction. It is arguable that bootstrapping is what fuels the explosion in language and conceptual development that occurs around the third year of child development [28], [26].

There is also a link between action and language. [29] provided a generative grammar describing the structure of

action. This grammar has both computational applicability and a biological basis.

C. Conclusion

In the current paper we combined computer vision based action representation and classification with a bootstrapping process to accelerate (non-linguistic) acquisition of action knowledge in a robot. As discussed above, structural bootstrapping performs a comparison of the meaning (semantics) of actions at the level of SECs and – if required – then in a second step a comparison of its individual syntactic elements (descriptors d). This way it becomes for the first time possible to perform the rather complex aspects of Accommodation and Assimilation [9] in a formal and algorithmically sound way with a robot-compatible action encoding. The resulting categorization allows for a better understanding of the underlying actions and their cognitive meanings. In [10] we demonstrate that the here-introduced action representation can be used to execute the respective action with a robot. Thus, learning the representation from observation together with robot execution does - we think - provide a substantial contribution to the field of cognitive robotics

References

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

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

[3] M. Pardowitz, S. Knoop, R. Dillmann, and 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*, vol. 37, no. 2, pp. 322–332, 2007.

[4] S. Ekvall and D. Kragic, "Robot learning from demonstration: a task-level planning approach," *International Journal of Advanced Robotic Systems*, vol. 5, no. 3, pp. 223–234, 2008.

[5] R. Cubek and 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, J. Dörr, K. Ning, B. Dellen, and F. Wörgötter, "Learning the semantics of object-action relations by observation," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1229–1249, 2011.

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

[8] E. E. Aksoy, B. Dellen, M. Tamosiunaite, and 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] J. Piaget, *The Origins of Intelligence in the Child*. London, New York: Routledge, 1953.

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

[11] A. Abramov, K. Pauwels, J. Papon, F. Wörgötter, and B. Dellen, "Real-time segmentation of stereo videos on a portable system with a mobile GPU," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1292–1305, 2012.

[12] A. Abramov, E. E. Aksoy, J. Dörr, K. Pauwels, F. Wörgötter, and 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.

[13] T. Kulvicius, K. J. Ning, M. Tamosiunaite, and F. Wörgötter, "Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 145–157, 2011.

[14] J. A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proc. 2002 IEEE Int. Conf. Robotics and Automation*, 2002, pp. 1398–1403.

[15] S. Schaal, P. Mohajerian, and A. Ijspeert, "Dynamics systems vs. optimal control–a unifying view," *Prog. Brain Res.*, vol. 165, pp. 425–445, 2007.

[16] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance," in *Proc. 2009 IEEE Int. Conf. Robotics and Automation*, 2009, pp. 1534–1539.

[17] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 800–815, 2010.

[18] F. Wörgötter, E. E. Aksoy, N. Krüger, J. Piater, A. Ude, and M. Tamosiunaite, "A simple ontology of manipulation actions based on hand-object relations," *IEEE Transactions on Autonomous Mental Development, (In press)*, 2012.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, nov 2004.

[20] N. Badler, "Temporal scene analysis: Conceptual descriptions of object movements," Ph.D. dissertation, University of Toronto, Canada, 1975.

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

[22] H. Kjellström, J. Romero, and D. Kragić, "Visual object-action recognition: Inferring object affordances from human demonstration," *Comput. Vis. Image Underst.*, vol. 115, no. 1, pp. 81–90, jan 2011.

[23] Y. Yang, C. Fermuller, and Y. Aloimonos, "Detection of manipulation action consequences (mac)," in *International Conference on Computer Vision and Pattern Recognition (CVPR), (In press)*, 2013.

[24] K. Ikeuchi and T. Suehiro, "Toward an assembly plan from observation, part I: Task recognition with polyhedral objects," *IEEE Trans. Robotics and Automation*, vol. 10, no. 3, pp. 368–385, June 1994.

[25] S. Pinker, *Language Learnability and Language Development*. Cambridge: Cambridge University Press, 1984.

[26] J. Trueswell and L. Gleitman, "Learning to parse and its implications for language acquisition," in *Oxford Handbook of Psycholinguistics*, Oxford, 2007, pp. 635–656.

[27] G. Chierchia, "Syntactic bootstrapping and the acquisition of noun meanings: the mass-count issue," in *Heads, Projections and Learnability Volume 1*, B. Lust and J. W. MArgarita Suner, Eds. Hillsdale, New jersey, 1994, pp. 301–318.

[28] F. Tracy, "The language of childhood," *Am. J. Psychol.*, vol. 6, no. 1, pp. l07–138, 1893.

[29] K. Pastra and Y. Aloimonos, "The minimalist grammar of action," *Philosophical Transactions of the Royal Society B*, vol. 367, pp. 103–117, 2012.