Prediction of Manipulation Action Classes Using Semantic Spatial Reasoning

Fatemeh Ziaeetabar[1], Tomas Kulvicius[1], Minija Tamosiunaite[1,2] and Florentin Wörgötter[1]

***Abstract*— Human-robot interaction strongly benefits from fast, predictive action recognition. For us this is relatively easy but difficult for a robot. To address this problem, here we present a novel prediction algorithm for manipulation action classes in natural scenes. Manipulations are first represented using the Enriched Semantic Event Chain (ESEC) framework. This creates a temporal sequence of static and dynamic spatial relations between the objects that take part in the manipulation by which an action can be quickly recognized. We measured performance on 32 ideal as well as real manipulations and compared our method also against a state of the art trajectory-based HMM method for action recognition. We observe that manipulations can be correctly predicted after only (on average) 45% of action's total time and that we are almost twice as fast as the HMM-based method. Finally, we demonstrate the advantage of this framework in a simple robot demonstration comparing two different approaches.**

I. Introduction

In most cases, action recognition is considered as a classification problem, mapping image sequences to previously known actions. In general, here the question arises "how fast" can an action be recognized. Many systems will only respond *after* an action has finished, while here we are concerned with action prediction, leading to a system that provides recognition output *before* an action has completed. This is also the way humans interpret actions performed by others: we continuously perceive and update our belief about an ongoing action not waiting for its end.

Many applications exist, where action (or event) prediction is beneficial in autonomous navigation, surveillance, health care, and others. Two examples can make this clear: 1) driver action prediction to prevent accidents or 2) prediction of a handicapped person's looming fall and a proactive help by a robot. While in these two examples post-hoc recognition will usually not help, action prediction may prevent problems.

For a robot, the capability of on-line prediction (and behavioral adaptation) in a human-robot interaction scenario is a difficult and challenging problem, because human actions are complex, performed in variable ways [1], and decisions must be made based on incomplete action executions [2]. In this work, we are interested in manipulation action-class prediction. If one wants to analyse (and/or predict) the dynamics of an action, fully continuous action information — for example hand trajectories — should be used. For action-class prediction, this is not needed. Instead, here we focus on very simple hand-object and object-object relations, like "getting closer", "moving together", etc. The strength of this approach is that we only have to use a very small set of such relations to achieve high predictive power. To achieve this, in the current study we extend our recently introduced action classification framework based on Enriched Semantic Event Chains (ESECs) [3] to implement temporal action prediction. Each action is distinguished and classified semantically "as fast as possible" according to the differences in static and dynamic spatial information between the involved objects. We show with different experiments that this creates a new and robust framework for real time action prediction.

II. Related Work

There has been a great deal of research in the field of human activity recognition from simple human actions in constrained situations [4][5][6][7] to complex actions in cluttered scenes or in realistic videos [8][9][10][11]. Also there are recent works in early event detection that have attempted to expand human action recognition towards action prediction [12][13][14][15][16]. These approaches try to predict actions from incomplete video data.

Ryoo [12] proposed a method which explains each activity as an integral histogram of spatio-temporal features. Their recognition methodology named dynamic bag-of-words considers sequential nature of human activities and uses those for prediction of ongoing activities.

Cao et al. [13] proposed an optimization approach and formulated the problem of action prediction as a posterior maximization problem. They randomly removed some frames in a video to simulate missing data and then performed feature reconstruction based on previous frames for creating new frames. After that, the accuracy of the newly created features are computed by comparing them to those in the actual next frames.

Kong et al. in [2] proposed a structured SVM learning method to simultaneously consider both local and global temporal dynamics of human actions for action prediction. In another study [14] it had been proposed to use a deep sequential context network (DeepSCN), which first elegantly gains sequential context information from full videos and then uses the resulting discriminative power to classify partial videos.

∗The research leading to these results has received funding from the DFG grant WO 388/13-1 and the EU Horizon 2020 research and innovation program under grant agreement No. 680431, ReconCell.

[1]Fatemeh Ziaeetabar, Tomas Kulvicius, Minija Tamosiunaite and Florentin Wörgötter are with III. Physics Institute, University of Göttingen, Friedrich-Hund-Platz 1, 37077 Göttingen, Germany fziaeetabar@gwdg.de

[2]Minija Tamosiunaite is also with Faculty of Informatics, Vytautas Magnus University, Lithuania.

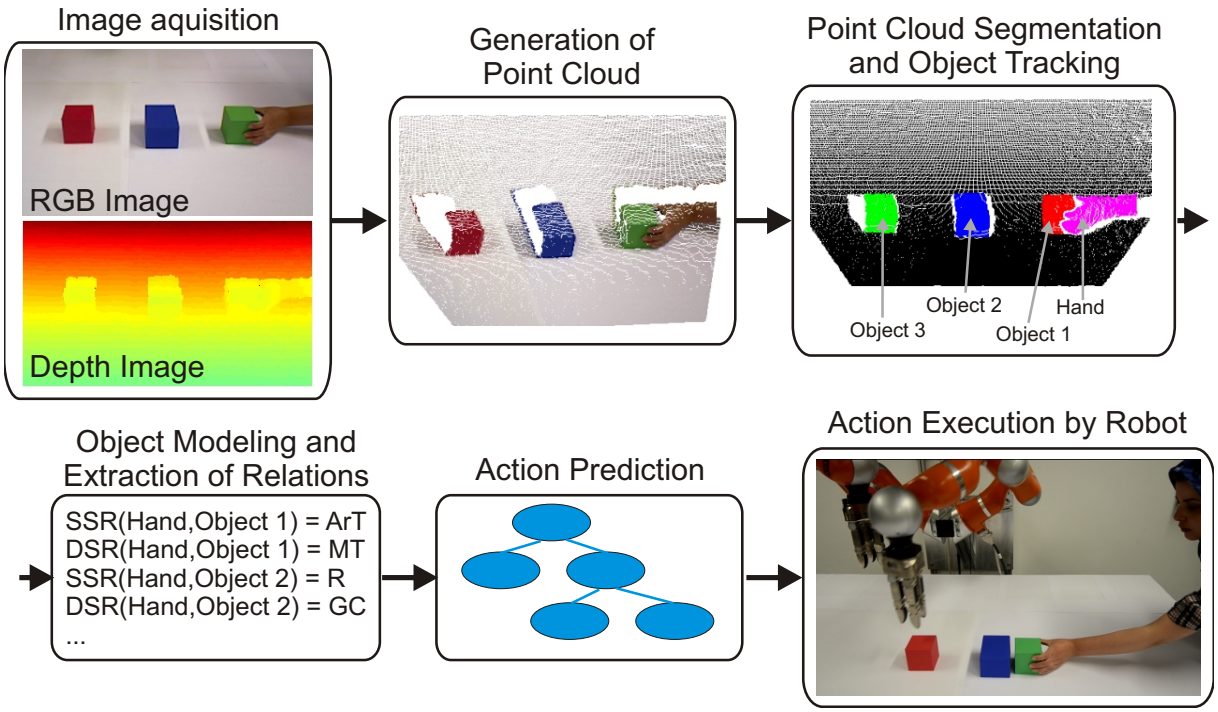


Fig. 1. Flow diagram of the prediction algorithm including human-robot interaction.

The importance of action prediction has been demonstrated recently in several robotic applications [15] [16]. For example [15] anticipates future activities from RGB-D data by considering human-object interaction. This method has been embedded into a real robot system to interact with a human in regular daily tasks. It considers each possible future activity using an anticipatory temporal conditional random field (ATCRF) that models the rich spatial-temporal relations through object affordances and then considers each ATCRF as a particle, and represents the distribution over the potential future activities using a set of particles. In our approach, we do not use particle filters; instead we represent each action as a matrix of spatial relations. Wang et al. [16] used probabilistic modelling of human movements for intention inference and action prediction. They applied an Intention-Driven Dynamics Model (IDDM) as a latent variable model for inferring unknown human intentions and performed predictions according to that.

In another work about prediction in human-robot interaction, a joint assembly task is specified and provided by a finite state machine representation. Here the robot learns to predict the next action of the human by discovering repeated patterns of low level actions like grasping an object. By assuming that repeated low level actions also imply repeated higher level sub-tasks, the robot learns to predict human actions [12]. This low-to-high level transfer may, however, often not really hold. A more sophisticated state/action model is described in [13], who applied an adaptive Markov model to assign confidence regarding predictions of the human partners' actions.

Our focus in the current work is on manipulations, which are actions performed by hands. Recently Fermüller et al. have developed a recurrent neural network based method for manipulation action prediction [17]. They depicted the hand movements before and after contact with the objects during the preparation and execution of actions and applied a recurrent neural network (RNN) based method while patches around the hand were their input. They additionally used the estimations of forces on finger tips during the different manipulations for having more accurate predictions.

A central problem that can be found in all of the above approaches is that action recognition (and prediction) heavily relies on time-continuous information (e.g. trajectories, movie sequences, etc.). This type of information, however, is highly variable. It is interesting to note that — indeed — we (humans) have a hard time to describe an action in words using this level of detailed-ness. Instead, we prefer using relational descriptions like "X moves toward Y", or "X is on top of Y". We may add "... is moved fast..." or similar specifiers but we usually cannot express in words detailed information on the actual speed, etc. Therefore, in this study we decided to shy away from continuous descriptions, too, trying to obtain leverage from a relational representation as discussed in our older works [18][19][20], which makes this system robust against individual spatial and temporal variations in the actual action execution. We will continue to discuss these issues in the Conclusion section, arguing that time-continuous information (dynamics) may not play much of a role for action-class prediction.

III. Overview of Our Method

First we will explain the whole process and then its components. A workflow diagram of action prediction and

execution is shown in Fig. 1. For each video frame, RGB and depth images are used to generate point clouds. Next, a segmentation algorithm based on color and depth information is used for preprocessing the input to extract and track objects and the hand in a scene using algorithms presented in [19] and [21]. Since segmentation and tracking is not the main focus of the current work, we will not discuss those methods in more detail.

Note that for action recognition, the ESEC framework used here [3] does not require any object and movement recognition. It only considers the spatial relations between objects. Since objects have different sizes and shapes we need to model them as simpler structures for judging their spatial relations. For this we use "Axis Aligned Bounding Boxes" (AABB).

Static and Dynamic spatial relations (SSR and DSR) are then computed according to the relative positions of these bounding boxes (for details see section IV-B). After that we define the Enriched Semantic Event Chain (ESEC) framework in section IV-C. An ESEC represents an action based on the relative spatial relations between the objects in a scene. Whenever a spatial relation changes, the corresponding change-event is stored in a transition matrix, the "ESEC".

The temporal action prediction is then formalized in section IV-E. The prediction algorithm is a step by step procedure that utilizes the ESEC matrices in order to discriminate actions according to their event chains.

Results are then analyzed or, in case of a robotic experiment, used to trigger the robot action.

For quantifications, we used the MANIAC data set [19] [1]. This data set consists of the following eight manipulation actions: push, put, take, stir, cut, chop, hide and uncover. Each action type is performed in 15 different versions by five human actors. Each version has a differently configured scene with different objects and poses.

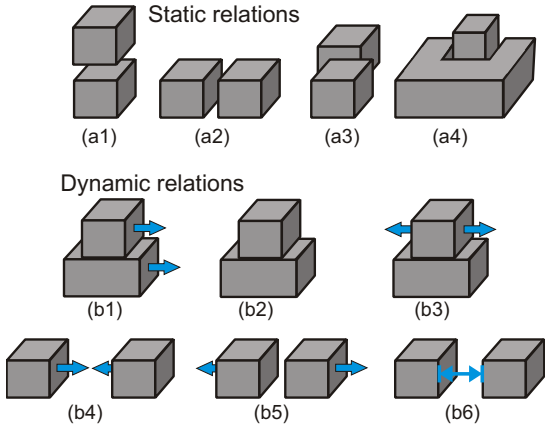
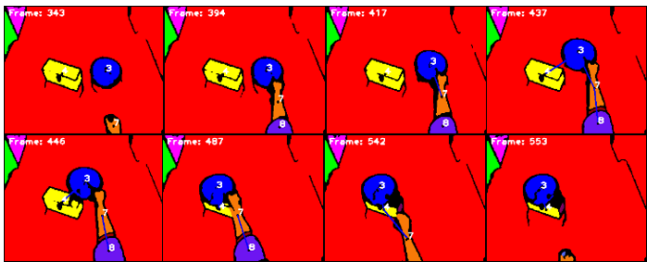


Fig. 2. (a) Static Spatial Relations: (a1) Above/Below, (a2) Right/Left, (a3) Front/Behind, (a4) Around. (b) Dynamic Spatial Relations: (b1) Moving Together, (b2) Halting Together, (b3) Fixed-Moving Together, (b4) Getting Close, (b5) Moving Apart, (b6) Stable.

[1]Publicly available at: http://www.dpi.physik.uni-goettingen.de/cns/index.php?page=maniac-data set.



H,M	U	T	T	T	T	T	T	N	N	N
M,P	U	U	U	N	N	N	N	N	N	N
M,S	U	U	U	U	U	U	T	T	T	T
P,S	U	U	U	U	U	U	T	T	T	T
H,P	U	U	U	N	N	N	N	N	N	N
H,S	U	U	U	U	U	U	N	N	N	N
H,M	U	Ab	To	To	To	To	To	To	Ab	O
M,P	U	U	U	Ab	Ab	Ab	Ar	Ar	Ar	Ar
M,S	U	U	U	U	U	U	Ab	To	To	To
P,S	U	U	U	U	U	U	Ab	To	To	To
H,P	U	U	U	Ab	Ab	Ab	Ar	Ar	Ar	O
H,S	U	U	U	U	U	U	Ab	Ab	Ab	O
H,M	U	HT	HT	MT	MT	MT	HT	MA	MA	Q
M,P	U	U	U	MA	S	GC	S	S	S	S
M,S	U	U	U	U	U	U	HT	HT	HT	HT
P,S	U	U	U	U	U	U	HT	HT	HT	HT
H,P	U	U	U	MA	S	GC	S	S	MA	Q
H,S	U	U	U	U	U	U	S	S	S	Q
	1	2	3	4	5	6	7	8	9	10

Fig. 3. Description of "Put on Top" action in SEC and ESEC frameworks. Image frames (top): Frame segmentation of a "Put on Top" video. Blue object (3) is the main object (M), table is the primary object (P) and yellow object (4) is the secondary object (S). Event matrix (bottom): White cells of the table - SEC matrix; green cells - ESEC Static relation matrix; blue cells - ESEC dynamic spatial relation matrix. The ESEC framework uses the whole table, while the SEC framework only includes the white part.

IV. Detailed Methods

A. Object Modelling

After segmentation, each object in a scene is represented as a point cloud that includes a set of points in a three dimensional coordinate system. Our scene at frame f is defined as a set of point clouds: $\alpha_1^f, ..., \alpha_N^f$, where N is the number of objects and α_i represents the point cloud of object i, which is tracked throughout the action-sequence [3]. We approximate each point cloud as an Axis Aligned Bounding Box (AABB) to allow for efficient detection of spatial relations. An AABB is a model that surrounds a point cloud by a box such that its sides are parallel to the directions of the axes of the coordinate system.

B. Extraction of Spatial Relations

In this work, three types of spatial relations have been considered: 1) "Touching" (**T**) and "Non-touching" (**N**), 2) Static Spatial Relations (**SSR**) and 3) Dynamic Spatial Relations (**DSR**) [3]. T and N relations between two point clouds of objects are determined by applying the "kd-tree algorithm" and evaluating occurrence (or non-occurrence) of collision between them [22].

Both static and dynamic spatial relations between two objects can be extracted simultaneously by evaluating the relations between AABBs of the objects. In the following, we will describe SSR and DSR in more detail.

1) Static Spatial Relations: Static spatial relations rely on the relative position of two objects in space. They do not need any data from previous frames and determine relations only at the current time moment (frame).

We define the following types of SSRs: "Above" (**Ab**), "Below" (**Be**), "Right" (**R**), "Left" (**L**), "Front" (**F**), "Back"(**Ba**) and "Between" (**Bw**). Right, Left, Front and Back are merged into "Around" (**AR**) or "Not-Around" (**N-Ar**) if one object is surrounded by the other or not, respectively. Moreover, "Above", "Below" and "Around" relations in combination with "Touching" are converted to "Top" (**To**), "Bottom" (**Bo**) and "Touching Around" (**ArT**), respectively, which correspond to the same cases but now with physical contact.

If two objects are far from each other or they have not any of the above mentioned relations, their static relation is assumed as Null (**O**). This leads to a set of 12 static relations: ***SSR*** = {Ab, Be, To, Bo, R, L, F, Ba, Ar, ArT, N-Ar, O}.

Fig. 2 (a1-a4) represents static spatial relations between two objects in terms of cubes.

2) Dynamic Spatial Relations: Dynamic spatial relations define the spatial relation of two objects during movement of either or both of them. Here, different from *SSR*, some information from the previous K frames (e.g., distance related parameters) between each pair of objects is necessary.

The parameter K is related to the frame-rate of the movie, where we determine K as frame number for covering 0.5 seconds, which is a good estimate for the time that a human takes to change the relations between objects. Therefore, if the video rate is μ frames per second, then $K = 0.5\mu$.

DSRs consist of the following relations: "Moving Together" (**MT**), "Halting Together" (**HT**), "Fixed-Moving Together" (**FMT**), "Getting Close" (**GC**), "Moving Apart" (**MA**) and "Stable" (**S**). Dynamic spatial relations between two objects in term of cubes are shown in Fig. 2 (b1-b6). MT, HT and FMT denote situations when two objects are touching each other while both of them are moving in a same way (MT), are constant (HT), or one object is fixed and does not move, while the other one is moving on or across it (FMT). Case **S** denotes that any distance-change between objects is less than a defined threshold (here, we have considered this threshold as $\xi = 1$ cm) and remains constant during the action sequence. The other cases are clear from looking at Fig. 2 (b). In addition, **Q** is used to denote a dynamic relation between two objects if their distance is more than the defined threshold ξ or if they have not any of the above defined dynamic relations.

Thus, we have a set of seven dynamic relations:
DSR = {MT, HT, FMT, GC, MA, S, Q}.

C. Action Representation by ESEC

The ESEC framework is inspired by the original Semantic Event Chain (SEC) framework [1]. The original SECs consider only touching (T) and not-touching (N) events between all pairs of objects along a manipulation action and focus on the changes of these relations (see white rows of the matrix in Fig. 3). Here (U) annotates the situation that the role of the respective fundamental object is not yet known. The definition of object roles is given in Table I. (Note, objects *obtain* their role through the course of the action!). The extracted sequences of relational changes had been used for recognition of manipulation actions. In the Enriched SEC (ESEC) framework, in addition to touching and not-touching relations, sequences of static and dynamic relations described in section IV-B are analyzed (see green and blue rows of the matrix in Fig. 3).

TABLE I
DEFINITION OF THE FUNDAMENTAL OBJECTS DURING A MANIPULATION ACTION [3].

Object	Definition	Relation
Hand	The object that performs the action.	Not touching anything at the beginning and at the end of the action. It touches at least one object during an action.
Main	The object which is directly in contact with the hand.	Not touching the hand at the beginning and at the end of the action. It touches the hand at least once during an action.
Primary	The object from which the main object separates.	Initially touches the main object. Changes its relation to not touching during an action.
Secondary	The object to which the main object joins.	Initially does not touch the main object. Changes its relation to touching during an action.

It is important to note that one does not have to extract all relations between each pair of objects in a scene. It is only necessary to consider the so-called "fundamental objects", which are those that have an essential role in the manipulation for determining an ESEC matrix. This has been discussed in [3] and is an important step forward for reducing action-analysis complexity. This way, we naturally exclude distractor objects without any role in our manipulation and reduce computations.

D. Manipulation Action Ontology [20]

Manipulations can be divided into three main groups (Fig. 4 A): "Hand-Only Actions", "Separation Actions" and "Release Determined Actions". *Hand-Only Actions* are actions where the hand alone acts on a target object (or first grasps a tool and then the tool acts on the target object). According to their goals and effects on the scene they can be subdivided into "Rearranging" (like stirring) and "Destroying" (like cutting) actions. *Separation Actions* denote actions where the hand manipulates one object to either destroy it or remove it from another object. This group is also divided into two cases: "Break" (ripping-off) and "Take-Down" (taking down one object from another one). Finally, there are so-called *Release Determined Actions*, which include all actions where the hand manipulates an object and combines it with another object. This type of actions is subdivided into "Hide" (covering an object with another one) and "Construct" (building a tower) [20]. According to this

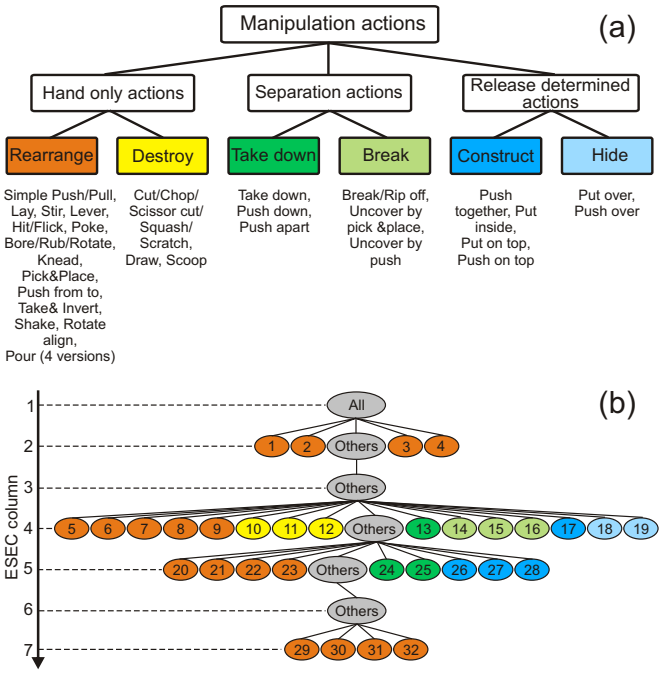


Fig. 4. (a) Categorization of 32 manipulation actions. (b) Prediction tree of manipulation actions according to ESEC framework.

subdivision, in this work, we have analyzed and categorized 32 manipulation actions as listed in Fig. 4 (a).

E. Action Prediction and Quantification Measures

We define these 32 actions as $\alpha_1, \alpha_2, ..., \alpha_{32}$. Each action in the ESEC framework has its own matrix with a specific total number of columns N_i $(1 \leq i \leq 32)$. For the theoretical analysis the event chains for all 32 actions were manually created in an ideal and noise free way. Furthermore, α_i^k denotes the *k-th* column of action α_i. Due to the predefined set of fundamental objects, the number of rows is 18 and is the same for all actions.

The distinct structure of the ESECs allows for temporal action prediction, which can be shown as a tree diagram (Fig. 4 (b)). This will be discussed in the Results section.

We call the column number in a SEC or ESEC at which the prediction of an action has occurred the "Prediction Event Column". This parameter for action α_i is displayed as $E(\alpha_i)$. We define a *prediction power* measure for the event based prediction as below (in percent):

$$P_E(\alpha_i) = \left(1 - \frac{E(\alpha_i)}{N_i}\right) * 100\%. \quad (1)$$

Hence, here the completion of an action corresponds to 1. A prediction power of 0% would then correspond to the case where action recognition only happens at the very end of the action while 100% would refer to the action's start.

Due to noise that exists in real data (e.g., due to inaccuracies in segmentation, detection of object collisions, etc.), predictions using real data will often not correspond exactly to theoretical predictions. Thus, we define another prediction power measure for the "frame based" evaluation. In this case, the spatial relations of the objects involved are computed for each video frame. The frame, at which the prediction occurs, is called "Prediction Frame".This parameter for action α_i is displayed as $F(\alpha_i)$. Similarly, prediction power for the frame based prediction is defined as below:

$$P_F(\alpha_i) = \left(1 - \frac{F(\alpha_i)}{L(\alpha_i)}\right) * 100\%, \quad (2)$$

where $L(\alpha_i) = lastframe(\alpha_i) - firstframe(\alpha_i)$, is the total number of frames during execution of action α_i and denotes the length of the action. We assumed as the first frame the one where the hand appears in the scene and the frame where the hand leaves the scene is the last frame.

F. Method for Quantification against Baseline Method

To assess our method against the state of the art, we compared our results with the performance of a state of the art HMM-based baseline from [23] applied on the MANIAC data set. For a fair comparison we selected this method, because—like ours—it does not use object information, but, instead, relies on hand trajectories.

We use the hand gesture recognition method from [23] for detection of the hand motions and then extend recognition to prediction. In [23] detection and segmentation of a hand takes place using 3D depth maps and color information. Then the hand trajectory is quantized based on an *orientation* feature, which provides the direction of motion between consecutive trajectory points of the hand. This extracted feature is clustered to generate discrete vectors, which are used as input to the HMMs recognizer and then the gesture path is classified using these discrete vectors. Evaluation, Decoding and Training as the main problems of an HMM model are solved by using Forward or Backward algorithm, Viterbi algorithm and the Baum-Welch algorithm respectively as in [23]. We adopted the same procedures here, too.

G. Methods for Human-Robot Interaction Experiments

The goal of this part of the work is to show that earlier action prediction leads to a benefit in cooperation. To this end, we have chosen two quite simple, but illustrative cases for human-robot interaction: 1) push blocks together and 2) put one block on top of the other block. In this study, we are not interested in complex computer vision and, therefore, we kept the scenario minimal. It just consists of a table with three coloured blocks as shown in Fig. 1. The human performs an action (push together or put on top); the robot observes this and is supposed to engage in the same action as soon as possible. Experiments were done comparing both SEC and ESEC approaches. For this, we used a KUKA LWR robotic-arm (see Fig. 1; in our experiments only one of the arms was used) and an ASUS-Xtion RGB-D sensor for getting the input data for the action prediction system.

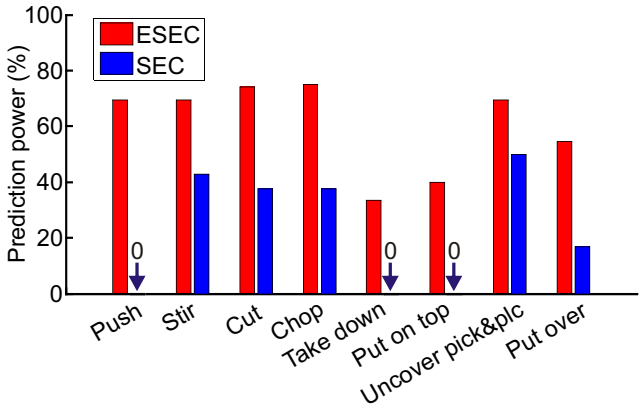


Fig. 5. SEC vs. ESEC in theoretical prediction analysis ($P_E(\alpha_i)$) on MANIAC data set actions.

We used the *Library of Manipulation Actions* proposed by [20] in order to generate motions and execute actions by the robot.

V. Results

We have compared the performance of action prediction using the ESEC against SEC and HMM frameworks on three different cases: 1) theoretical prediction of actions, 2) action prediction using the MANIAC data set, and 3) real robot experiments.

A. Prediction of Manipulation Actions

1) Theoretical Analysis of All Actions: For this comparison, we manually generated 32 ideal matrices for the representation of manipulation actions (see Fig. 4 (a), small print at the bottom) based on ESEC sequences, as explained above.

First we show how action prediction evolves over time. For this we build a decision tree (Fig. 4 (b)) as follows: At the start of an action, all first columns of the 32 manipulations $\alpha_i^1, (1 \leq i \leq 32)$ are compared. Then, all actions with the same first column are categorized into the same set $(S_1, ..., S_n)$. Afterwards, the members of each set are compared according to their second column α_i^2. Again, those actions with the same second column are categorized into the same set and this process is continued until all actions are categorized into a single-member set where there are no more identical columns with any of the other actions or all the columns of an action have been analysed.

The resulting tree uses the same color code as in Fig. 4 (a) and shows that maximally seven columns in an ESEC are needed until all actions are recognized. Note, the most complex action ("pouring") has in total 16 columns. Columns 1, 3, and 6 have no added discriminative value. Four actions are found already in column 2, where the bulk is discriminated in columns 4 and 5. Different action types (color code) are distributed along the tree and no type clustering is observed.

To quantify this better, we used the Prediction Event Column for each action and computed the prediction power $P_E(\alpha_i)$ for all 32 actions for both SEC and ESEC. We obtained an average prediction power of **18.10%** (SD=16.3%) when using the SEC framework and **52.68%** (SD=13.2%) for ESEC. This means that we can predict actions using ESECs much earlier (before half of the action has been completed) than when using SECs. Moreover, all of those 32 manipulation actions were recognized correctly when using ESEC, whereas only 20 actions out of 32 were recognized correctly when using SEC.

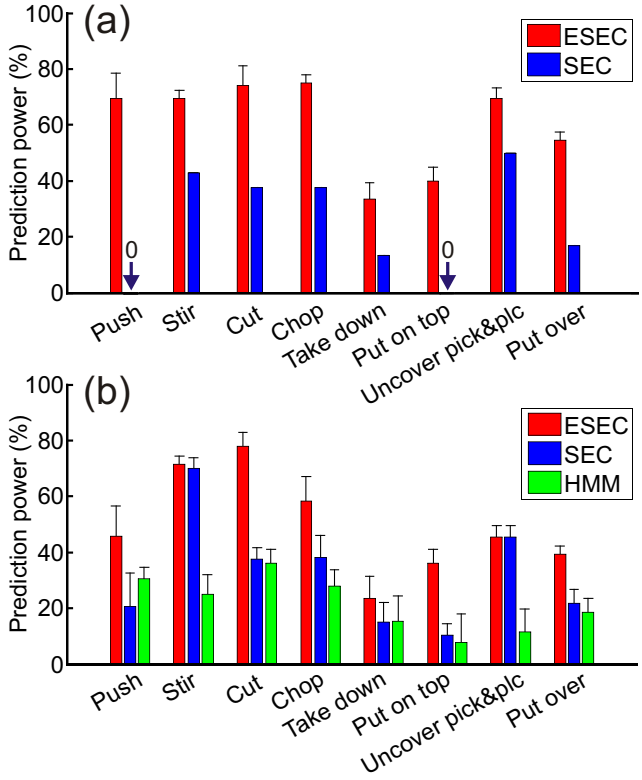


Fig. 6. Results of the comparison of action prediction using SEC and ESEC as well as the HMM methods on the MANIAC actions. (a) Event-based prediction ($P_E(\alpha_i)$). (b) Frame-based prediction ($P_F(\alpha_i)$). The error bars show standard deviations.

2) Theoretical Analysis of MANIAC-Type Actions: A comparison of the theoretical prediction power between SEC and ESEC for only the actions contained in the MANIAC data set is shown in Fig. 5. MANIAC-type action had for this been re-created in a noise-free manner. The average of the theoretical (best possible) prediction power for MANIAC-type actions is **23.1%** (SD=21.2%) for SEC and **59.8%** (SD=15.5%) for ESEC.

TABLE II

Comparison of prediction power for SEC and ESEC obtained from theoretical analysis (MANIAC-type actions), and SEC, ESEC and HMM on MANIAC data set. Average and standard deviation is shown.

	Theory	**MANIAC**	
	P_E	P_E	P_F
SEC	23.1%±21.2%	24.7%±19.1%	32.3±19.3%
ESEC	59.8%±15.5%	60.7±15.5%	51.3%±17.9%
HMM	n/a	n/a	21.6%±18.5%

B. Action Prediction on MANIAC Data Set

To see how well theory matches to reality, we performed the same analysis now using the real MANIAC movies [2]. We have randomly selected three versions of each of the existing eight actions, thus, here we used 24 actions in total. We have calculated and compared both prediction power measures, i.e., "Event based" ($P_E(\alpha_i)$) and "Frame based" ($P_F(\alpha_i)$).

Results for the comparisons between prediction powers of MANIAC manipulations between SEC and ESEC frameworks and an HMM-based method as a baseline method are presented in Fig. 6. Here, panel (a) shows Event-based prediction and panel (b) denotes frame-based prediction. Values in Fig. 6 (a) slightly differ from Fig. 5 because of some inaccuracies in computations of spatial relations and presence of noise in real data. In most of the cases ESECs predict faster than SECs and the HMM-based method in all cases. This is confirmed by Table II, which shows the average prediction power for all eight manipulations of the MANIAC data set for both event- and frame-based evaluations. ESECs are on average **36%** better than SECs in event-based and **19%** better in frame-based real data analysis. Moreover, ESECs are totally **29.7%** better than HMM-based method in frame-based prediction of MANIAC manipulation actions. Furthermore, the ESEC method is of lower algorithmic complexity than the HMM-based one.

In general, comparing all panels show that all different (theoretical and real-data) analyses lead to consistent results.

C. Action Prediction in Robot Experiments

One of the most promising applications of the proposed prediction method concerns human-robot or a robot-robot interaction. By using our prediction method, a robot can anticipate a human's or another robot's action before the action has ended and engage in collaboration as soon as the action is predicted. To demonstrate this, as explained above we designed and performed two robot experiments: "Push together" and "Put on top". Here, the task for the robot was to observe the human action and then engage in a collaboration by performing the same action as soon as the action is recognized.

Using ESECs, a put on top action is predicted when the hand and the main object (green block) are getting close to the secondary object (blue block), whereas with SECs, this action is predicted only after the hand places the main object (green block) on the secondary object (blue block) and releases it (an un-touch event is detected; see also supplementary video). For the push together action, the ESEC predicts the action at the moment when the hand starts moving together with the main object (green block), whereas when using a SEC, the action is predicted only after the hand pushes the main object (green block) toward the secondary object (blue block) and releases the main object. For these two manipulation actions, when using SECs a correct prediction is made very much at the end of these actions (prediction power of **15.4%** for **push together** action and **9.1%** for **put on top** action), whereas when using ESECs, predictions can be made much earlier (**45.5%** and **23.8%**, respectively).

We show selected frames from these robot experiments in Fig. 7, where we can observe differences between prediction times (the frame when the robot predicted the action and started executing that action) for the push together and put on top actions when using the ESEC and SEC approaches. In case of the push together action, using SECs, the robot starts approaching the red block when the hand leaves the scene, whereas when using ESECs the robot has already completed the push together action and is moving back to the initial position (see elliptic marks on the frames). Similarly, in case of predicting a put on top action using SECs, the robot starts moving towards the red object when the action is already finished by the person and the hand leaves the scene, whereas in case of ESECs, the robot has by then already grasped the red object and lifted it up. Thus, as expected from the other analyses, in real robot experiments ESECs performed faster than SECs with a **30.1%** and **14.7%** improvement with ESEC in comparison to SEC for **push together** and **put on top** actions, respectively.

VI. Conclusion

In this paper, we proposed an approach to manipulation action prediction based on the ESEC framework and compared it with SEC and an "object-free" HMM-based method. We showed that on overage the ESEC framework outperforms both SEC and HMM-based methods. One possible strength of ESEC (and SEC) is that is does not rely on time-continuous information, which—in all likelihood—is far more prone to variability (and noise) than the *quasi-symbolic* representations used by ESEC (and SEC). Indeed, when watching some of the examples in the MANIAC data set one sees that time continuous information will not improve prediction much, because the only aspect added by this is the action dynamics. Dynamics do not influence the action *class* but will play a role in the way *how* an action is executed (e.g. fast versus slow, etc.). This, however, is irrelevant for manipulation action-class prediction. Furthermore, our prediction approach as opposed to [14][15][16] does not need any action trajectories, shape features or action reconstruction and performs prediction only by using semantic representation and spatial relations in a simple way. This has low complexity, can perform in real time scenarios and is strongly linked to the way human language describes an action.

References

[1] J. Dinerstein, D. Ventura, and P. K. Egbert, "Fast and robust incremental action prediction for interactive agents," *Computational Intelligence*, vol. 21, no. 1, pp. 90–110, 2005.

[2] Y. Kong, Z. Tao, and Y. Fu, "Deep sequential context networks for action prediction," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pp. 1473–1481, IEEE, 2012.

[3] F. Ziaeetabar, E. E. Aksoy, F. Wörgötter, and M. Tamosiunaite, "Semantic analysis of manipulation actions using spatial relations," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 4612–4619, IEEE, 2017.

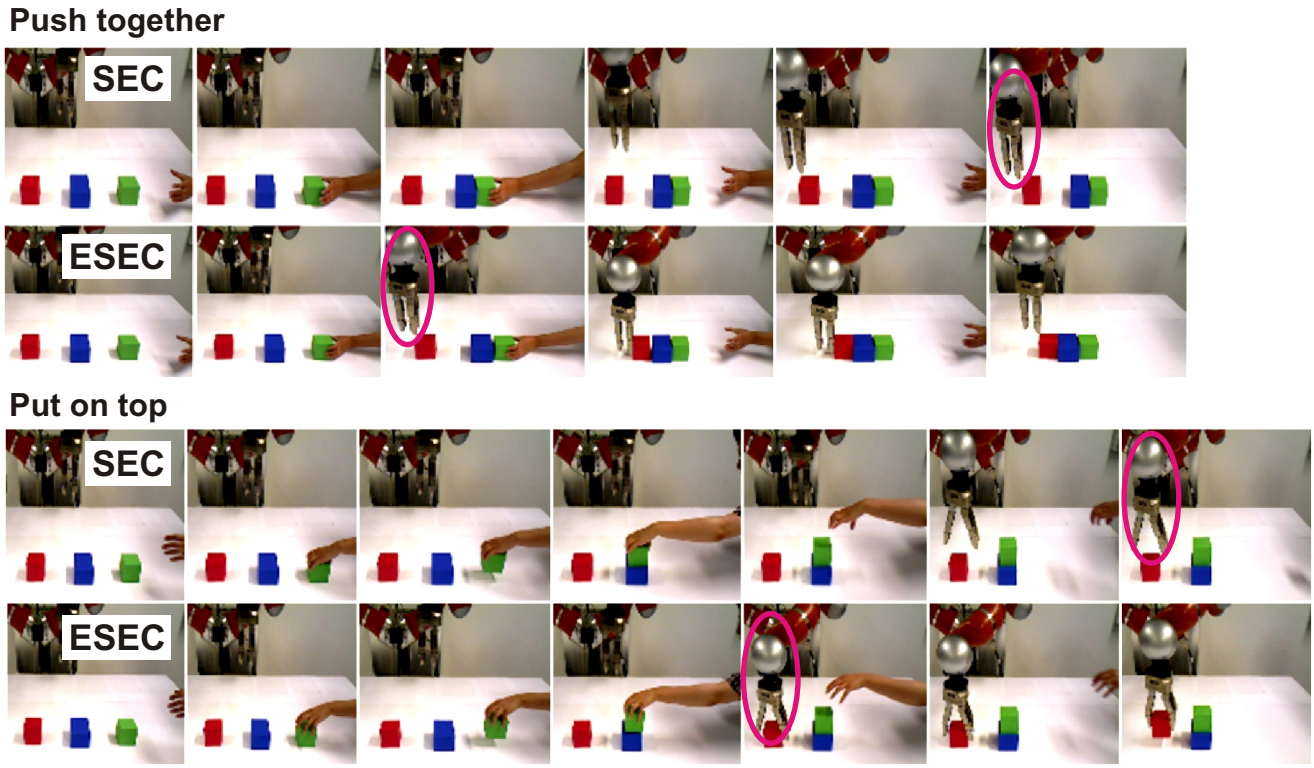


Fig. 7. Results from robot experiments. Ellipses mark stages at which the robot engages into an action. For more details please refer to the main text.

[4] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," in *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pp. 90–102, IEEE, 1997.

[5] J. W. Davis and A. F. Bobick, "The representation and recognition of human movement using temporal templates," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 928–934, IEEE, 1997.

[6] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer vision and image understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[7] C. Wallraven, B. Caputo, and A. Graf, "Recognition with local features: the kernel recipe," in *null*, p. 257, IEEE, 2003.

[8] T. Lan, L. Sigal, and G. Mori, "Social roles in hierarchical models for human activity recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1354–1361, IEEE, 2012.

[9] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[10] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman, "Structured learning of human interactions in tv shows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2441–2453, 2012.

[11] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1331–1338, IEEE, 2011.

[12] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1036–1043, IEEE, 2011.

[13] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. Mark Siskind, and S. Wang, "Recognize human activities from partially observed videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2658–2665, 2013.

[14] T. Lan, T.-C. Chen, and S. Savarese, "A hierarchical representation for future action prediction," in *European Conference on Computer Vision*, pp. 689–704, Springer, 2014.

[15] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 14–29, 2016.

[16] Z. Wang, M. P. Deisenroth, H. B. Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic modeling of human movements for intention inference," *Proceedings of robotics: Science and systems, VIII*, 2012.

[17] C. Fermüller, F. Wang, Y. Yang, K. Zampogiannis, Y. Zhang, F. Barranco, and M. Pfeiffer, "Prediction of manipulation actions," *International Journal of Computer Vision*, pp. 1–17, 2016.

[18] 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.

[19] E. E. Aksoy, M. Tamosiunaite, and F. Wörgötter, "Model-free incremental learning of the semantics of manipulation actions," *Robotics and Autonomous Systems*, vol. 71, pp. 118–133, 2015.

[20] 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*, vol. 5, no. 2, pp. 117–134, 2013.

[21] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter, "Voxel cloud connectivity segmentation-supervoxels for point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2027–2034, 2013.

[22] M. Aiello and B. Ottens, "The mathematical morpho-logical view on reasoning about space," in *IJCAI*, pp. 205–211, 2007.

[23] M. Elmezain, A. Al-Hamadi, and B. Michaelis. "Hand gesture recognition based on combined features extraction," World Academy of Science, Engineering and Technology 60 (2009): 395.