GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Spatio-temporal reasoning for semantic scene understanding and its application in recognition and prediction of manipulation actions in image sequences

Dissertation
ZUR ERLANGUNG DES MATHEMATISCH-NATURWISSENSCHAFTLICHEN DOKTORGRADES "DOKTOR RERUM NATURALIUM" DER GEORG-AUGUST UNIVERSITÄT GÖTTINGEN

im Promotionsprogramm PCS
der Georg-August University School of Science (GAUSS)

vorgelegt von
Fatemeh Ziaeetabar
aus Teheran, Iran

Göttingen 2019

Thesis Committee:

First Supervisor: Prof. Dr. Florentin Wörgötter
Second Supervisor: Prof. Dr. Dieter Hogrefe

Members of the examination board:

First Reviewer: **Prof. Dr. Florentin Wörgötter**,
Georg-August-Universitt Gttingen, Faculty of Physics, Third Institute of Physics

Second Reviewer: **Prof. Dr. Minija Tamosiunaite**,
Vytautas Magnus University, Faculty of Informatics, Department of Systems' Analysis

Other Members of the examination board:

Prof. Dr. Marcus Baum,
Georg-August-Universität Göttingen, Faculty of Mathematics and Computer Science, Institute of Computer Science

Prof. Dr. Carsten Damm,
Georg-August-Universität Göttingen, Faculty of Mathematics and Computer Science, Institute of Computer Science

Prof. Dr. Dieter Hogrefe,
Georg-August-Universität Göttingen, Faculty of Mathematics and Computer Science, Institute of Computer Science

Prof. Dr. Wolfgang May,
Georg-August-Universität Göttingen, Faculty of Mathematics and Computer Science, Institute of Computer Science

I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules.

Göttingen 2019

Abstract

Human activity understanding has attracted much attention in recent years, due to a key role in a wide range of applications and devices, such as human- computer interfaces, visual surveillance, video indexing, intelligent humanoid robots, ambient intelligence and more. Of particular relevance, performing manipulation actions has a significant importance due to its enormous use, especially for service, as well as industrial robots. These robots strongly benefit from a fast and predictive recognition of manipulation actions. Although, for us as humans performing these actions is a quite trivial function, however this is not necessarily the case for a robot. To address this problem, in this thesis, we propose an approach for the representation, as well as an algorithm for the recognition and prediction of manipulation action categories, as observed in videos.

The key contributions of this thesis are the following: First, we modeled each object as a simple axis aligned bounding box and provide a qualitative spatial reasoning method to calculate static and dynamic spatial relationships, accordingly. Static relations depend on the relative spatial position of two objects, including "Above", "Below", "Right", "Left", "Front", "Back", "Inside", "Surround", "Around without touch", "Around with touch", "Top" and "Bottom"; while dynamic relations address the spatial relation of two objects during movement of either or both of them. These relations consist of "Getting close", "Moving apart", "Stable", "Moving together", "Halting together" and "Fixed moving together". This qualitative approach allows us to provide a new semantic representation of manipulation actions, creating a sequence of static and dynamic spatial relations between the manipulated objects taking part in a manipulation. Our approach creates a transition matrix, called the "Enriched Semantic Event Chain (ESEC)". The rows of this matrix show spatio-temporal relations include touching/ not-touching (rows 1:10), static (rows 11:20) and dynamic (rows 21:30) relations within each pair of manipulated objects, while the columns of the matrix contain events that occur as a result of one or more change(s) in the spatio-temporal relations between the involved objects. Since the presence of noise as well as inappropriate accuracy in object modeling may lead to errors in the calculation of spatio-temporal relations, our framework has been adapted to the algorithm of noise identification and correction.

Second, we designed clustering and classification algorithms according to the ESEC framework, to distinguish and recognize manipulation actions. To this end, we introduced a novel method to calculate the similarity between manipulation actions. Our algorithm is validated on a data-set including 120 scenarios of 8 action types obtaining an accuracy of 95%.

Third, the ESEC framework is employed to predict a large set of manipulations in theoretical as well as real data . Our method could correctly predict manipulation actions after only (on average) 45% of their execution was accomplished, which is twice as fast as a standard Hidden Markov Model based method. This claim, was tested on 35 theoretically defined manipulations as well as two publicly available data-sets consisting of a total of 162 scenarios in 12 action types.

Finally, we designed a cognitive experiment to examine the prediction of manipulation actions in a virtual reality-based environment. To this end, we selected 10 actions distributed in all possible groups and subgroups of manipulations. Next, we designed and created 300 scenarios of these

actions, producing a large data-set of manipulation actions in a virtual reality environment. To our knowledge, this is the first virtual reality data-set of human manipulation actions, aimed at helping AI scientists studying human action recognition. In the next step, we performed an experiment where 50 human subjects participated in, and were asked to predict the type of action in each scenario, before it ends. Our ESEC-based prediction method was applied on these scenarios, proving capable of predicting the manipulation actions as good as 17.6% faster than the human participants.

The main advantage of our proposed framework, ESEC, is that it is capable of encoding a manipulation in a highly invariant and abstract way, independent from object poses, perspectives and trajectories which could largely interchange. In fact, ESECs help resolve the problem of action representation under conditions where clutter and big scenes induce complexities in the analysis of scaled matrices.

Different from model-based policy designs, our model-free framework operates on spatio-temporal object relations without making assumptions on the structure of objects and scenes. This new form of representation, enables us to provide the novel recognition and prediction algorithms for manipulation actions, leading to a high efficiency.

Acknowledgments

The work included in this thesis could not have been possible without terrific expert support. First of all, I would like to thank my supervisors Prof. Dr. Florentin Wörgötter and Prof. Dr. Minija Tamosiunaite for guiding me through my thesis by sharing their valuable experiences with me and for the countless hours of fruitful discussions without which this work could not have been accomplished. Further appreciation goes to Prof. Dr. Ricarda Schubotz, Dr. Eren Erdal Aksoy and Dr. Tomas Kulvicius, whom I was lucky to receive expert advice from throughout the thesis.

I would also like to thank all my former and current colleagues for their direct or indirect input to my work and for having great time together. Many thanks go to Aisha Aamir, Dr. Mohamad Javad Aein, Dr. Alejandro Agostini, Johannes Auth, Moritz Becker, Dr. Mayte Bonilla Quintana, Dr. Jan-Matthias Braun, Dr. Michael Fauth, Dr. Juliane Herpich, Sebastian Herzog, Dr. Tatyana Ivanovska, Dr. David Kappel, Jannik Luboeinski, Timo Lüddecke, Dr. Daniel Miner, Dr. Timo Nachstedt, Dr. Jeremie Papon, Stefan Pfeiffer, Dr. Simon Reich, Dr. Jan Markus Schoeler, Florian Teich, Dr. Christian Tetzlaff and Erenus Yildiz. Special thanks to Ursula Hahn-Wörgötter who was always a big help.

I am honored to express my sincere gratitude to the members of the examination board, Prof. Dr. Dieter Hogrefe, Prof. Dr. Marcus Baum, Prof. Dr. Wolfgang May and Prof. Dr. Carsten Damm.

I deeply appreciate my parents Ali and Ashraf, the first sources of encouragement in my scientific life, for their continuous support, wise guidance and inspiration to follow my dreams restlessly. I thank my brother and sister, Mohsen and Maryam for all the joyful moments together.

Last but far from least, I appreciate my husband Dr. Moein Esghaei, for his endless love, support and dedication. He represents the most important motivation and inspiration during the challenges of PhD and life. I really do not have words to describe the deeply love that I feel for him.

Fatemeh Ziaeetabar
Göttingen 2019

Dedicated to my husband, Moein
my mother, Ashraf
and my father, Ali

List of related publications

Journal paper:

(a) **Ziaeetabar, F.**, Kulvicius, T., Tamosiunaite, M., & Wörgötter, F., "Recognition and Prediction of Manipulation Actions Using Enriched Semantic Event Chains", Robotics and Autonomous Systems (RAS), vol. 110, pp. 173-188, 2018.

(b) Wörgötter, F., **Ziaeetabar, F.**, Pfeiffer, S., Kaya, O. Kaya & T., Tamosiunaite, M., "Humans Predict Action using Grammar-like Structures", Scientific Reports 10.1 (2020): 1-11.

(c) **Ziaeetabar, F.**, Pomp, J., Pfeiffer, S., El-Sourani, N., Shubotz R.I., Tamosiunaite, M., & Wörgötter, F., "Human and Machine Action Prediction Independent of Object Information", Submitted to: Nature Human Behaviour (NHB).

Conference papers:

(d) **Ziaeetabar, F.**, Pfeiffer, S.,Tamosiunaite, & Wörgötter, F., "Anticipation of Everyday Life Manipulation Actions in Virtual Reality", 2019 IEEE Conference on Signal Image Technology and Internet based Systems (SITIS), Italy, 2019 (in press).

(e) **Ziaeetabar, F.**, Pfeiffer, S.,Tamosiunaite, M., Kulvicius, T., & Wörgötter, F., "Who Can Predict Faster? Human or Robot ", 2019 Anticipation and Anticipatory Systems: Humans Meet Artificial Intelligence (CREA), Sweden, 2019.

(f) **Ziaeetabar, F.**, Kulvicius, T., Tamosiunaite, M., & Wörgötter, F., "Prediction of manipulation action classes using semantic spatial reasoning", 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3350-3357, IEEE, 2018.

(g) **Ziaeetabar, F.**, Aksoy, E.E., Tamosiunaite, M., & Wörgötter, F., "Semantic analysis of

manipulation actions using spatial relations", 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 4612-4619, IEEE, 2017.

Extended abstracts:

(h) **Ziaeetabar, F.**, Tamosiunaite, M., Wörgötter, F., "A Novel Semantic Framework for Anticipation of Manipulation Actions", at Anticipating Human Behavior Workshop in European Conference on Machine Vision (ECCV), Munich, Germany, 2018.

(i) **Ziaeetabar, F.**, Wörgötter, F., Aksoy, E.E., "Extraction of Spatial Object Relations for Understanding the Semantics of Manipulation Actions", at Semantic Policy and Action Representations (SPAR) Workshop in IEEE International Conference on Robotics and Intelligent Systems (IROS), Hamburg, Germany, 2015.

Dataset:

We provided a large "Manipulation Action dataset in Virtual Reality Environment" (**MAVRE**) with 10 different manipulations including (chop, Cut, Hide, Uncover, Put on top, Take down, Lay, Push, Shake and Stir). Each of which consists of 30 different versions (totally 300 scenarios) performed by 2 different human actors. All objects, including hand and tools, in all actions are represented by colored cubes of variable size, color, and location. All manipulations were recorded with the Vive virtual reality headset and controller.

This data-set was made in a collaboration with Stefan Pfeiffer.

The **MAVRE** has been introduced and used in (b) and (c).

Contents

List of Figures

List of Abbreviations

AABB	**A**xis **A**ligned **B**ounding **B**ox
CFG	**C**ontext **F**ree **G**rammar
DMP	**D**ynamic **M**ovement **P**rimitive
DSR	**D**ynamic **S**patial **R**elation
DTW	**D**ynamic **T**ime **W**arping
ESEC	**E**nriched **S**emantic **E**vent **C**hain
HMM	**H**idden **M**arkov **M**odel
LCS	**L**ongest **C**ommon **S**ub-sequence
LfD	**L**earning from **D**emonstration
MAVRE	**M**anipulation **A**ctions **V**irtual **R**eality **E**nvironment
OBB	**O**riented **B**ounding **B**ox
NLP	**N**atural **L**anguage **P**rocessing
PCFG	**P**robabilistic **C**ontext **F**ree **G**rammar
QSR	**Q**ualitative **S**patial **R**easoning
SEC	**S**emantic **E**vent **C**hain
SR	**S**patial **R**easoning
SSR	**S**tatic **S**patial **R**elation
VR	**V**irtual **R**eality

Chapter 1

Introduction

1.0.1 Motivation

One of the central goals in cognitive robotics is to analyze, recognize and predict human behaviors. The large application of this topic in the field of computer vision and robotics confirms its major role in human-human, as well as human-robot interactions. Key applications of cognitive robotics fall into the following categories:

- **Industrial**
  Industrial service robots can be used to carry out a wide range of tasks, from simple, such as examining welding spots, to complicated and harsh-environment cases, such as aiding in dismantling nuclear power stations [1].
- **Frontline Service Robots**
  Service robots are system-based autonomous and adaptable interfaces that interact, communicate and deliver services to an organization's customers [2].
- **Domestic**
  Domestic robots perform tasks that humans regularly perform in non- industrial environments, such as housework, including cleaning floors, mowing the lawn and pool maintenance [3]. People with disabilities, as well as elder people, may soon access such service robots to help them live independently [3].
- **Scientific**
  Autonomous scientific robots perform tasks which are hardly possible for humans, from missions from deep in the ocean, to those in outer space [4].

While most of the researches in the field of "Human Activity Analysis" focus on full-body action categorization, one major requirement for a service robot is the ability to manipulate objects found in human environments. However, almost all the robots developed by experts in AI and robotics,

per- form poorly in manipulating objects and executing tasks compared to even non-skilled human (ex., a child). The manipulation ability of humans is because of their excellent brain processing capabilities together with the high performance sensors (eyes) and flexible actuators (hands), while a robot or an intelligent system needs a lot of factors to obtain this ability. In all manual interactions a robot makes with humans and the environment, it must be able to identify the scene together with the spatial relations between manipulated objects, determine the type of actions and produce an appropriate response.

On the other hand, although human activity recognition is beneficial for some offline analysis, however it fails to be enough in lots of real time applications. In real world applications, such as autonomous navigation, surveillance systems, health care, etc., post-hoc recognition is usually not helpful and we need to predicatively recognize actions early in time to prevent problems.

For a service robot, the capability of on-line prediction (and behavioral adaptation) in a human-robot interaction scenario is a difficult and challenging problem, because human manipulation actions are complex, performed in variable ways, and decisions must be made based on incomplete action executions.
This thesis' contributions span from the area of representation to recognition and prediction of manipulation actions. Our specific goals are summarized as follows:

- To employ spatial reasoning techniques to calculate static and dynamic spatial relations between objects in a scene space.
- To define a semantic framework for definition and representation of manipulation actions according to the spatial relations.
- To develop a manipulation actions prediction algorithm which uses enriched semantic event chains in a hierarchical tree structure for distinguishing between different types of manipulations.
- To integrate the designed recognition and prediction semantic frame- work with virtual reality and compare the prediction results with human performance, as well as the existing mathematical prediction algorithms.

The remaining of this chapter is organized as follows: The problems we address are stated in 1.2. A review of the state-of-the-art techniques concerning our approach is provided in 1.3. The contributions of this thesis are summarized in 1.4 to conclude this chapter.

1.0.2 Problem Statement

Spatial Reasoning

Semantical scene understanding involves the assessment of the spatial arrangement of objects. Using spatial relations not only helps us discriminate the objects in the scene [5], but also allows us to distinguish between different interpretations of two scenes with similar objects with different spatial arrangements [6]. Spatial relations are abstract and functional relationships between entities in space which can create a new perspective on action identification.

Here, we aim to present a manipulation action recognition and prediction framework which does not use object recognition information and represents manipulations in terms of spatial relations between their manipulated objects. To develop a theory of spatial relations, it is necessary to determine the minimal set of spatial relations needed to describe the spatial organization of objects.

Here, in order to facilitate the computation of spatial relations, we use the camera axes and create a simple Axis Aligned Bounding Box (AABB) surrounding each object and perform calculations based on the relationships between the AABBs.

Spatial relations are divided into *static* and *dynamic* relations.

Static Spatial Relations (SSR) depend on the relative position of two objects in space and include "Above", "Below", "Right", "Left", "Front", "Back", "Inside" and "Surround". Right, Left, Front and Back are merged into "Around". The relations "Above", "Below" and "Around" are assumed to happen in case the relation "Not touching" holds. When paired with the "Touching" relation (that is, two objects are in physical contact with each other), the corresponding relations are called: "Top", "Bottom" and "Touching Around".

Dynamic Spatial Relations (DSR) 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. Dynamic relations consist of "Getting close", "Moving apart", "Stable", "Moving together", "Halting together" and "Fixed moving together".

Manipulation Actions Representation and Recognition

There have been two main approaches to this problem based on symbolic and geometric (sub-symbolic) representations. The symbolic approach is most common within classic AI and natural language communities. Engineers and roboticists usually prefer more geometric approaches dealing with low-level signals.

Both approaches have their pros and cons. The symbolic approach is more intuitive in tasks related to understanding and communication with humans. It also generates a discrete state space which makes planning tasks more tractable compared with the signal space which is of continuous nature. However, the major problem is the grounding of these symbols in the environment. In signal space the main problem is to find a small subset of features for manipulation actions. Two demonstrations of the same pick and place action could look totally different in signals space, which makes it difficult to find a conjoined symbolic representation for this action [7]. More recent approaches, including our approach, try to combine both approaches to have the benefits of both. Our proposed framework, named as *Enriched Semantic Event Chains (ESECs)* creates a temporal sequence of static and dynamic spatial relations between the objects that take part in the manipulation action. Mathematically speaking, ESECs are transition matrices that symbolically encode the relational static and dynamic changes between (unspecified) objects. Each row of an ESEC matrix represents the sequence of the spatial relations between each pair of manipulated objects attained during the continuous video. Whenever a change occurs in any of those spatial relations a new column is created. As a consequence, each column reflects at least one such change.

After a proper action representation, action recognition is implemented by comparing the action ESEC matrix of a new action (test sample) to the action ESECs matrices of existing action models (training samples) and computing the similarity score. We assign the class label to the tested action as the one belonging to the action, which had the maximal similarity score.

Ontology of Manipulation Actions

Humans can robustly classify objects and actions using a very high degree of invariance and generalization. To reach such a high classification robustness in artificial systems, we created a large ontology of manipulation actions by taking ESECs as reference. This helps to understand how manipulation actions are fundamentally structured in the spatio-temporal domain.

Manipulation Actions Prediction

We humans constantly update our believes about both ongoing actions and future events. We easily recognize on- going actions, but there is even more to this. We can understand the kinematics of the ongoing action, the limbs' future positions and velocities. We also understand the observed actions in terms of our own motor-representations. That is, we are able to interpret others' actions in terms of dynamics and forces and predict the effects of these forces on objects. Similarly, cognitive robots that assist human partners need to understand their intended actions at an early stage. If a robot needs to act, it cannot have a long delay in visual processing. It needs to recognize in real-time to plan its actions. A fully functional perception-action loop requires the robot to predict, so it can efficiently allocate future processes. Finally, even vision processes for multimedia tasks may benefit from being predictive [8].

In this thesis, we are specifically interested in manipulation actions and how visual information of hand and manipulated objects can be exploited for predicting future actions. Here, the special way of manipulation actions representation in ESEC method by using static and dynamic spatial relations allows us to use the ESEC action matrices for action prediction. For this, the Touching or Not touching relation(T/N), Static Spatial Relation (SSR), and Dynamic Spatial Relation(DSR) are computed for each pair of so called "fundamental" objects. We consider the object to belong to the set of fundamental objects if this object is being touched or untouched by some other object during the action. For action prediction, we perform column-wise comparison of the matrix of that action to the matrices from the training data set (in this case we use several action matrices as models for each action class) until all actions are categorized into a set which consists of the action members from the same class, or where there are no identical columns with any of the actions. In the latter case, we compute the similarity measure as presented later for those incomplete action tables and predict the label based on the maximum similarity score. If case scores are identical for several action from different classes, we proceed to the next column until a unique class is obtained.

Validation of Manipulations Prediction Method in Virtual Reality

Although our ESEC prediction algorithm has outstanding results in both theory and practice, however like any other scientific method, we need to validate it by comparing with the other existing methods. Therefore, it is necessary to design a suitable substrate for comparison. Consequently, we compared our semantic method with a state-of-the-art hand trajectory recognition algorithm according to Hidden Markov Model (HMM) [9] and [10] as a mathematical approach.
Next, we compared our algorithm's predictability power with humans. To this end, we selected 10 actions which are distributed in all possible groups and subgroups of manipulations, including Chop, Cut, Hide, Uncover, Put on top, Take down, Lay, Push, Shake, Stir and made 30 sample scenarios of each in Virtual Reality (VR) (totally 300 scenarios), each scenario with a different

geometrical and coloring setup. We next asked 50 individuals to join our VR experiments and do action prediction. Results were next com- pared with the result of ESEC method applied on exactly the same data.

1.0.3 State-of-the-art

For each of the problems mentioned in 1.2 a review of the existing literature will be presented.

Spatial Reasoning

Qualitative spatial and temporal reasoning is a sub field of knowledge representation and symbolic reasoning that deals with knowledge about an infinite spatio-temporal domain using a finite set of qualitative relations. One particular aim in this type of reasoning, is to model human common-sense understanding of space. Spatial relations as an aspect of spatial reasoning are used in many applications in various domains, in medical images to recognize different brain structures [11, 12], in image interpretation to provide linguistic scene descriptions [13], in Geographical Information Systems (GIS) applications to computer-aided design [14] and in robotics [15, 16]. Mobile robot navigation is an important topic in the field of spatial robots reasoning that involves "self-localization", "map learning" and "human-robot communication" issues. In self-organization, the location of the robot is determined based on spatial relations with respect to the perceived objects [17], and map learning involves the autonomous acquiring of the environment's map [18,19]. Moreover, service robots are supposed to take orders from humans and, in some cases, report back to humans, or request more information to resolve ambiguities. In these scenarios, being able to communicate spatial information is a key capability [20, 21]. All these applications require a thorough analysis of space and spatial relations between entities.

1.0.4 Manipulation Actions Representation and recognition

Representation

There are two distinct approaches in action representation and executions. One at the trajectory level [22] and the other at the symbolic level [23]. The former gives more flexibility for the definition of actions, while the latter defines actions at a higher level which allows for generalization and planning actions at a higher level and allows for generalization and planning. For trajectory level representation there are several well established techniques, Splines [24], Hidden Markov Models (HMMs) [25], Gaussian Mixture Models (GMMs) [26] and dynamic Movement Primitives (DMPs) [22,27]. On the other hand, high level symbolic representations usually use graph structures and relational representations [28, 29]. Sridhar et al. [28] represented a whole video sequence by

an activity graph with discrete levels each of which represents qualitative spatial and temporal relations between objects involved in activities, however, large activity graphs and the difficulty of finding exact graph isomorphism make this framework expensive and sensitive to noise. Along the same line, Aksoy et al. [29] used semantic event chains (SECs) as a high level action descriptor. SECs are generic action descriptors that capture the underlying spatio-temporal structure of continuous actions by sampling only decisive key temporal points derived from the spatial interactions between hands and objects in the scene. In this thesis, we aim to improve SECs by adding static and dynamic spatial relations and define enriched semantic event chains (ESECs).

Recognition

Manipulation recognition can be understood as a sub-field within the above- discussed more general problem of human activity recognition. Numerous previous studies have attempted to solve this problem [28, 30–32].
To solve automatically recognize human manipulation activities from videos, Ramirez et al. suggested to extract functional object categories from spatio-temporal patterns encoding the interactions between hand and objects in a semantic layer. This coding system is then used to analyze manipulation actions, although it suffers from a lack of generality in the semantic rules generator [28]. Furthermore, the authors of [30] and [31] try to improve the semantic action rules generator by exploring a reasoning method, which extracts these rules via employing abstract hand movements with the object information and enhance the recognition of manipulation actions through spatio-temporal feature learning. They show that by introducing new capabilities to the reasoning engine, one could compute new relationships between objects and actions, to improve hand action recognition. However their proposed method still does not work efficiently for complex hand movements with unknown movement primitives.

Due to the limitations in the physical To solve automatically recognize human manipulation activities from videos, Ramirez et al. suggested to extract functional object categories from spatio-temporal patterns encoding the interactions between hand and objects in a semantic layer. This coding system is then used to analyze manipulation actions, although it suffers from a lack of generality in the semantic rules generator [28]. Furthermore, the authors of [30] and [31] try to improve the semantic action rules generator by exploring a reasoning method, which extracts these rules via employing abstract hand movements with the object information and enhance the recognition of manipulation actions through spatio-temporal feature learning. They show that by introducing new capabilities to the reasoning engine, one could compute new relationships between objects and actions, to improve hand action recognition. However their proposed method still does not work efficiently for complex hand movements with unknown movement primitives.

Due to the limitations in the physical modeling of movements, caused by the variation of action types and their components, researchers have developed graph-based approaches. In [33] visual

semantic graphs are introduced for recognition of manipulation sequences according to the changes in the topological structure of the manipulated objects. Another study modeled human manipulations by incorporating semantic information about human skeleton and tracking the segments of manipulated objects [34]. Faria et al. used hand trajectories and hand-object interactions in a Bayesian model to enable manipulation understanding. These methods share a drawback in that they are not efficient enough for complex and hybrid applications [35]. In order to solve the above drawback, Aksoy et al. described a method for semantic segmentation and recognition of long and complex manipulation actions, which captured the underlying spatio-temporal structure of an action and extracts the basic primitive elements of each parsed manipulation [32]. Building on this, a more descriptive set of spatial relations between manipulated objects were introduced in [36] (see also [37]) which can be lead to more precise action representation and recognition.

were introduced in [36] (see also [37]).

1.0.5 Manipulation Actions Prediction

Our focus in the current work is not only to recognize but also to rapidly predict manipulations. Recently, Fermüller et al. developed a recurrent neural network based method for manipulation action prediction [8]. They depicted the hand movements before and after contact with the objects during the preparation and execution of actions and applied a method based on a recurrent neural network (RNN) where patches around the hand were used as inputs to the network. They additionally used the estimations of forces on finger tips during the different manipulations to achieve more accurate predictions. Others [9,38] have used a hidden Markov model-based continuous gesture recognition system utilizing hand motion trajectories. We have here extended their methods from recognition to prediction and compared it with our ESEC approach [39].

A central problem that can be found in all of the above approaches is that action recognition (and prediction) heavily rely on time-continuous information (e.g. trajectories, movie sequences, etc.). This type of information, however, are highly variable. It is interesting to note that — indeed — we (humans) have a hard time describing 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 “... moves 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, as well, trying to obtain leverage from a relational representation as discussed in our older works [29,40,41], which makes this system robust against individual spatial and temporal variations in the actual action execution.

1.0.6 Virtual Reality

Virtual reality (VR) is a rapidly developing computer interface that strives to immerse the user completely within an experimental simulation environment, thereby providing a much more intuitive link between the computer and the human participants. VR has been applied successfully to hundreds of scenarios in diverse areas, including rapid prototyping [42], manufacturing [43], scientific visualization [44], engineering [45], and education [46]. Additionally, it has a considerable number of applications in the machine vision domain. Segmentation of 3D images, 3D shape modeling, 3D rigid and nonrigid registration, 3D motion analysis and 3D simulation are some important machine vision topics that can accurately match a virtual environment of graphically simulated 3D models to the video images of the real task environment [47]. Using three-dimensional(3D) images is becoming very popular in the medical research. This comes from the new capabilities demonstrated by computer vision applied to 3D imagery. Not only does it provide better diagnosis tools, but also new possibilities for therapy. This is true in particular for brain and skull surgery and radiotherapy, where simulation tools can be tested in advance, in a virtual environment, and next be used during the intervention as guiding tools [48].

In this thesis, after definition of a novel semantic framework (ESEC) for representation, recognition and prediction of manipulation actions and comparing the results in theory as well as the real data with the state of the art mathematical methods, we next carry out a comparison between predictability powers of humans and the ESEC framework. For this purpose, we selected 10 actions and made 30 scenarios for each in virtual reality environment with different geometrical and coloring setups. Next, we asked 50 human participants to participate in this experiment and predict the action types as early as possible while observing the action being performed. Afterwards, the ESEC results were compared with the human results.

1.0.7 Overview and Contributions

The contribution of each chapter can be summarized as follows:

- **Chapter 2:** This chapter was published in [36] and contains spatial reasoning notions, such as calculating static and dynamic spatial relations. Also, it includes basic concepts like object modeling, object roles, fundamental manipulated objects definition and uses these thoughts to provide a framework for semantic representation of manipulation actions. This framework is called Enriched Semantic Event Chain (ESEC)and is applied for recognition of manipulation actions in this chapter.
- **Chapter 3:** was published in [49] and introduces manipulation actions ontology and also categorization. It further presents a method for prediction of manipulation action classes using

spatial reasoning. Results are then used to trigger the robot action and we demonstrate the advantage of ESEC framework comparing two different approaches in a robotic experiment.

- **Chapter 4:** was published in [39] and includes a comprehensive definition of ESEC framework by using new object roles, new similarity measurement and a novel noise reduction algorithm. This framework is then used in recognition and prediction of manipulation actions in theory as well as real data. The results are obtained and discussed in two big data-sets. In the following, a state-of-the-art HMM based approach for recognition of manipulation is introduced and developed as a prediction method. Further, its results are compared with the ESEC results on both data-sets and the efficiency of the framework is evaluated.
- **Chapter 5:** provides describes the virtual reality system and the design of a VR-based experiment for action prediction and describes its aspects in a detailed manner. Afterwards, it reports human results in predicting the manipulation actions, analyzes them considering different aspects and compares the results of the ESEC framework applied on the same data.

Finally, in **Chapter 6** the thesis is concluded by a short summary and final remarks.

Chapter 2

Spatial Reasoning and its Application in Representation and Recognition of Manipulation Actions

This chapter contains an original manuscript, presenting our fundamental framework for the classification and recognition of manipulation actions. It includes the following:

- Object modeling using the Axis-Aligned Bounding Box (AABB) approach.
- Spatio-temporal reasoning and the division of spatial relations into static and dynamic, as well as the formal description and computation procedures.
- Definition of fundamental object roles in a manipulation.
- Creating Enriched Semantic Event Chain (ESEC) as a temporal sequence of static and dynamic spatial relations between the fundamental objects taking part in a manipulation.
- Introducing our method to measure the similarity of ESEC matrices.
- Action classification using the ESEC framework on a large set of actions.
- Action discrimination in the ESEC framework using theoretical analyses.

2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Semantic Analysis of Manipulation Actions Using Spatial Relations

Fatemeh Ziaeetabar[1], Eren Erdal Aksoy[2], Florentin Wörgötter[1], and Minija Tamosiunaite[1, 3]

Abstract— **Recognition of human manipulation actions together with the analysis and execution by a robot is an important issue. Also, perception of spatial relationships between objects is central to understanding the meaning of manipulation actions. Here we would like to merge these two notions and analyze manipulation actions using symbolic spatial relations between objects in the scene. Specifically, we define procedures for extraction of symbolic human-readable relations based on Axis Aligned Bounding Box object models and use sequences of those relations for action recognition from image sequences. Our framework is inspired by the so called Semantic Event Chain framework, which analyzes touching and un-touching events of different objects during the manipulation. However, our framework uses fourteen spatial relations instead of two. We show that our relational framework is able to differentiate between more manipulation actions than the original Semantic Event Chains. We quantitatively evaluate the method on the MANIAC dataset containing 120 videos of eight different manipulation actions and obtain 97% classification accuracy which is 12 % more as compared to the original Semantic Event Chains.**

Index Terms—**Spatial relations, manipulation actions, semantic analysis, action semantics, action classification.**

I. INTRODUCTION

Action recognition and human activity analysis are the most active and challenging domains in computer vision and robotics. They play an important role in human-human as well as human-robot interactions. Also, it has many other applications in different fields such as video surveillance systems or video retrieval. Most of the researches in this area focus on full-body action categorization [1] [2], but there are a lot of tasks that an agent (human or robot) performs only using his hands (i.e., manipulation actions). Manipulation actions make a big proportion of applications both in industrial and service robotics. Intelligent robots could use observation of manipulation actions for learning how to manipulate. However, there are many ways to perform a single manipulation and it would be very inefficient to store a large set of observed examples that is not easy to generalize. The paper addresses the problem of representing manipulations in a compact and efficient way. It describes actions in terms of changes of spatial relations in the scene, while ignoring the diversity of scenes, objects and small details in the trajectory for doing the same action.

Spatial relations are abstract and functional relationships between entities in space [3]. One way of representing them is in the way humans speak about space [4] [5], e.g. "Top", "Bottom" or "Above", "Below". A correct understanding of object-wise spatial relations for a given action is essential for a robot to perform an action successfully [6]. Suppose, we ask a robot to put some object on the top of the other object. For a successful execution, in addition to the recognition of those two objects, the robot should have knowledge about "Above" and "Top" relations. It should take the first object and rise it to the "Above" of the second object and then put it on the "Top" of it. Definition of a robot action through appropriate spatial relations would lead to an accurate and generalizable performance in the robot execution.

In this regard, we apply qualitative spatial reasoning to each object pair in the scene. We use camera axes and create an Axis Aligned Bounding Box (AABB) around of each object. In the AABB representation, all box sides are parallel to the directions of axes. Next, we evaluate static and dynamic spatial relations, where the static relations set includes "Touching", "Non-touching", "Above", "Below", "Around", "Top", "Bottom" and the dynamic relations set includes "Getting Close", "Moving Apart", "Move Together", "Stable" and "Halt Together" for all pairwise objects. We design heuristic rules for evaluation of those relations and track changes in those relations during continuous video-frames.

The computed relations are embedded into the so called "Enriched Semantic Event Chain" representation, which is the extension of the original Semantic Event Chain approach [7] developed to semantically compare and identify actions [8]. We benchmark the proposed approach for accuracy in action recognition based on the MANIAC dataset [8] that includes 8 different manipulation actions (overall 120 videos performed by three different actors). To address wider action variety, we also show that the Enriched Semantic Event Chains in principle can differentiate between more actions as compared to the original Semantic Event Chains based on a 26 action set presented in [9].

II. RELATED WORKS

There has been a great deal of research in the field of spatial representation and reasoning because of its multifaceted applications in robot planning and navigation [10], interpreting visual inputs [11], computer-aided design [12], cognitive science where models of spatial skills help to

[1] Institute for Physics 3- Biophysics and Bernstein Center for Computational Neuroscience, Georg August University, Göttingen, Germany (e-mails: {fziaeetabar, worgott}@gwdg.de, minija.tamosiunaite@phys.uni-goettingen.de), [2] Karlsruher Institute for Technologies (KIT), Karlsruhe, Germany (e-mail: eren.aksoy@kit.edu). [3] Faculty of Informatics, Vytautas Magnus University, Kaunas, Lithuania.

explain human performance [13], geographic information systems (GIS) [14], and understanding natural languages [15]. All of these cases need to represent and reason about spatial aspects of the world. Spatial reasoning is studied using both quantitative and qualitative approaches. According to [16], quantitative reasoning is the developed (human) ability to analyze quantitative information and to determine which skills and procedures can be applied to a particular problem to arrive at a solution while a qualitative approach creates non-numerical descriptions of physical systems and their behavior, preserving important behavioral properties and qualitative distinctions. Qualitative spatial reasoning (QSR) provides representational primitives and inference mechanisms about space. In fact, QSR aims at capturing human-level concepts of space by using finite sets of relations to model particular spatial aspects such as topology, orientation and distance while quantitative spatial models rely on numeric calculations. Here, we would like to apply a qualitative approach because it is closer to how humans represent and reason using commonsense knowledge. It can overcome the indeterminacy problems, by allowing inference from incomplete spatial knowledge and it also offers a compact representation that is supposed to enable complex decision tasks.

Spatial reasoning techniques in artificial intelligence attempt to emulate human reasoning during navigation and other spatial planning tasks. For example, [18] applies results of brain research to obtain geometrical factors or [19] suggests a model in the form of spatial templates and prototypes (both quantitative spatial reasoning). A method of performing qualitative spatial reasoning on robots is proposed in [20].

Robotics is a domain much influenced by methods of spatial reasoning. One of the key aspects which is needed to understand commands such as “go in front of the closet door", is the ability of reasoning about spatial directions in a qualitative manner. In other words, the robot needs to be able to reason about an object with respect to another object in a given reference frame [20]. Therefore, finding spatial relations between objects in a scene is fundamental in execution of tasks by robots. In this work, we limit our study on manipulation actions that define actions which are done by hands. Because of large variation of ways for performing manipulation actions and also many occlusions in the visual scenes, manipulation action recognition is still an open and challenging problem. Meanwhile, hand movements as such have been widely investigated, but for a slightly different purpose: hand gesture recognition, for human-computer interfaces or sign language recognition [21].

In this study we concentrate on analysis of manipulation actions via the relations of manipulated objects. Only a couple of studies exist doing this type of analysis. In [22] visual semantic graphs were introduced for recognition of action consequence according to the changes in the topological structure of the manipulated objects. The study presented in [23] represents an entire manipulation by an activity graph which holds spatiotemporal interaction between objects, however, the activity graph requires complicated processing for extraction of semantic level knowledge. The work in [24] modeled human activities by involving some information about human skeleton and tracking the segments of manipulated objects. The authors of [25] use hand trajectories and hand-object interactions in a Bayesian model for manipulation observation. All the studies mentioned above introduce representations which don’t abstract from multiple execution details, while we attempt to describe manipulation actions through abstract relations. The already mentioned “Semantic Event Chain” (SEC) approach [7] is introduced as a possible generic descriptor for manipulation actions, which encodes the sequence of spatio-temporal changes in relations between manipulated objects. But it only takes into account touching and not-touching relations and does not consider other spatial information, therefore it has limitations in action recognition, as well in its usability for guiding execution by a robot. Here we would like to extend the SEC framework by considering qualitative static and dynamic spatial relations between objects and make a novel more accurate framework for classification of manipulation actions based on symbolic spatial relations.

III. OUR APPROACH

A. Overview of our method

A brief description of the steps involved in our approach is provided in Fig.1 and the details will be discussed in the following sections.

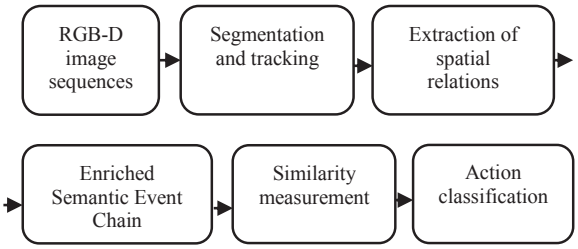


Fig. 1. Steps of our spatial reasoning approach

In order to semantically identify and compare manipulation actions, we present a new algorithm based on qualitative spatial relations. The input of our algorithm is an RGB-D video of a manipulation action. In this work, we use the videos of the MANIAC dataset which includes 8 different manipulation actions ***(Pushing, Hiding, Putting, Stirring, Cutting, Chopping, Taking, and Uncovering)*** [8].

A segmentation algorithm is applied on the scene at the first frame and objects are tracked during the rest of frames (section III-B). Spatial relations like “on top”, “above”, “below”, are extracted as described in section III-C and so called Enriched Semantic Event Chains (ESEC) are defined in section III-D. Finally, our similarity measures and classification procedure is described in section III-E. The discriminative ability of the ESECs for different actions is evaluated in section IV. Results are compared to analogous results obtained using the original Semantic Event Chains (SECs) as presented in [8, 9].

B. Point cloud segmentation and tracking

As the first step, the recorded video frames are pre-processed by an image segmentation procedure based on color and depth information as described in [8]. In this procedure objects (and hands) in the scene are extracted as separate

segments. A sample of a MANIAC dataset frame before and after segmentation is shown in Fig.2. Segments are tracked using a persistent super voxel world-model which is updated, rather than replaced, as new frames of data arrive as described in [26].

Each object in a scene after the aforementioned procedures is a point cloud, i.e., a set of points in a three-dimensional coordinate system (X, Y, Z). We define the scene at frame f as a set of point clouds: $\{\alpha_1^f,\ldots, \alpha_N^f\}$, where N is the number of objects in the f_{th} frame of the action. Object α_i^f represents the point cloud of object i at frame f, $i \in \{1,\ldots, N\}$ and can be tracked throughout the frames sequence.

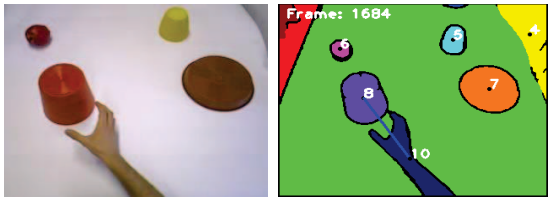


Fig. 2: A frame in MANIAC dataset (a) before and (b) after the scene segmentation. Segments are identified by different colors and segment numbers.

C. Extraction of spatial object relations

In this work, we define two types of spatial relations. The first type includes *static* relations which describe the directional ordering of objects in a scene and the second type contains *dynamic* relations between objects.

We define the following static spatial relations between objects in the scene: "Above" (**Ab**), "Below" (**Be**), "Right" (**R**), "Left" (**L**), "Front" (**F**), "Back" (**Ba**) and "Between" (**Bw**).

"**To**" and "**Bo**" explain top and bottom relations, respectively, which incorporate "Above" and "Below" with touching (**Ab** + **T** = **To**; **Be** + **T** = **Bo**). We gather all of these relations in a set and name it *Rel_static*. Thus, *Rel_static* = {**Ab**, **Be**, **R**, **L**, **F**, **Ba**, **Bw**, **To**, **Bo**}.

Dynamic relations are the second type of relations in the current study which are collected in a *Rel_dynamic* set. When an object starts moving and the distance between its central point and another object's central point decreases in a time interval they are "Getting Close" (**GC**) and when this distance increases, it means these two objects are "Moving Apart" (**MA**). We also observe "**MT** = Move Together" (here we mean only moving together when being in touching (**T**) relation), "**HT** = Halt Together" (touching but not moving) and "**S**=Stable" (non-touching (**N**), but keeping the same distance). Thus, *Rel_dynamic* = {**GC**, **MA**, **MT**, **HT**, **S**}. Note, the relations "Touching" (**T**) and "Non-touching" (**N**) making the backbone of the original Semantic Event Chain framework [7] are used in some of the definitions of our new relations (e.g. **To**, **Bo**, **MT**, **HT**, **S**) as described above.

Further we explain in more detail how the introduced relations are calculated in real scenes. The touching (**T**) and non-touching (**N**) relations are determined by applying the "kd-tree algorithm" on two point clouds [5] and evaluating occurrence (or non-occurrence) of collision between the point clouds.

For definition of the other relations we need to first introduce our object model. We define the coordinate axes according to the direction of the camera axes. Our coordinate system is shown in Fig.3. The z axis corresponds to perceived depth (front/back) direction, while the x and y axes define directions of right/left and above/below, respectively. Table 1 defines directions of six spatial relations in terms of the coordinate system axes.

For each point cloud (object) we create an Axis Aligned Bounding Box (AABB). In the AABB all sides are parallel to the directions of the coordinate system axes (Fig.3(b)).

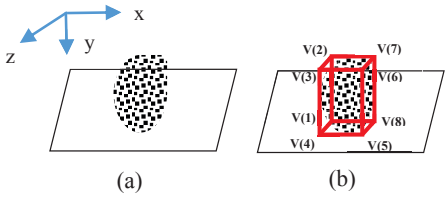


Fig.3. (a) Coordinate system, (b) A sample of AABB around a point cloud based in the defined coordinate system.

Suppose object α_i^f is the i_{th} object in the f_{th} frame represented as a point cloud and consisting of $P_{N\alpha i}$ points. As an object α_i^f model we define the AABB by the following set of vertices:

$$V_i^f(1) = [x_{min(i)}^f, y_{max(i)}^f, z_{min(i)}^f],$$
$$V_i^f(2) = [x_{min(i)}^f, y_{min(i)}^f, z_{min(i)}^f],$$
$$V_i^f(3) = [x_{min(i)}^f, y_{min(i)}^f, z_{max(i)}^f],$$
$$V_i^f(4) = [x_{min(i)}^f, y_{max(i)}^f, z_{max(i)}^f],$$
$$V_i^f(5) = [x_{max(i)}^f, y_{max(i)}^f, z_{max(i)}^f],$$
$$V_i^f(6) = [x_{max(i)}^f, y_{min(i)}^f, z_{max(i)}^f],$$
$$V_i^f(7) = [x_{max(i)}^f, y_{min(i)}^f, z_{min(i)}^f],$$
$$V_i^f(8) = [x_{max(i)}^f, y_{max(i)}^f, z_{min(i)}^f].$$

where $x_{min(i)}^f$, $x_{max(i)}^f$, $y_{min(i)}^f$, $y_{max(i)}^f$, $z_{min(i)}^f$ and $z_{max(j)}^f$ are the minimum and maximum values between the points of object α_i^f in the x, y and z axes, respectively. We calculate spatial relations only for objects which are "neighbors" in the scene where the neighborhood is defined in the following way: suppose O_i^f shows the central point of the AABB of object α_i^f; we define $\Omega(\alpha_i^f, \alpha_j^f) = \|O_i^f - O_j^f\|$ to be a two argument function for measuring the Euclidean distance between the objects α_i and α_j in f_{th} frame. Objects are considered to be neighbors in case $\Omega(\alpha_i^f, \alpha_j^f) \leq \tau$. In this study we define a threshold τ of 1 m, which makes most of the objects in our table-top manipulation neighbors (only extremely distant objects, e.g. those that are beyond the table are excluded).

Each relation is defined by a set of rules and those rules are evaluated for each neighboring object pair. We start with specifying the rules set for static spatial relations. Let us consider the relation "Right": $SR(\alpha_i^f, \alpha_j^f)$ = **R** (object α_i is to the right of object α_j in frame f) **if** $x_{max}(\alpha_i^f) > x_{max}(\alpha_j^f)$ as well as all the following (exception) conditions are *not* true:

$y_{min}(\alpha_i^f) > y_{max}(\alpha_j^f)$; $y_{min}(\alpha_j^f) > y_{max}(\alpha_i^f)$; $z_{min}(\alpha_i^f) > z_{max}(\alpha_j^f)$; $z_{min}(\alpha_j^f) > z_{max}(\alpha_i^f)$. The exception conditions exclude from the relation "Right" those cases when two object-AABBs do not overlap in altitude (y direction) or front/back (z direction). Several examples of objects holding relation *SR* (red, blue) = **R**, when the size and shift in y direction varies, are shown in in Fig. 4.

$SR(\alpha_i^f, \alpha_j^f) = \mathbf{L}$ is defined by $x_{min}(\alpha_i^f) < x_{min}(\alpha_j^f)$ and the same set of exception conditions. The relations "**Ab**", "**Be**", "**F**", "**Ba**" are defined in an analogous way. For "**Ab**" and "**Be**" the emphasis is on the "y" dimension, while for the **F**", "**Ba**" the emphasis is on the "z" dimension.

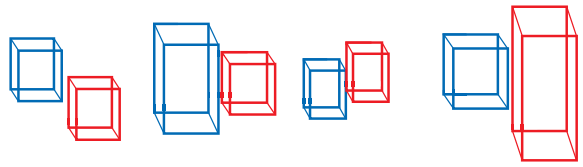


Fig.4. Possible states of Right- Left relations between two AABBs when size and y positions vary.

Next we will define the "**Bw**" (Between) relation (see Fig 5). First we define so called "Between space" for two objects. This space is obtained by extending the AABBs of two non-overlapping objects towards each other along the pre-defined axis and taking the intersection of those extensions. Whenever the third object's AABB completely stays in the "Between space" of the two other objects', it is assumed that the third object is in "Between" (**Bw**) of the two objects. The rules for this relation in the case the "Between space" is on the *X* axis is defined below (the object α_3^f is in between of objects α_1^f and α_2^f):

$$SR(\alpha_1^f, \alpha_2^f, \alpha_3^f) = \mathbf{Bw},$$
$$\mathbf{If}\ (x_{min(3)}^f > \text{maximum}\ (x_{min(1)}^f, x_{min(2)}^f)\ \&\&$$
$$(x_{max(3)}^f < \text{minimum}\ (x_{max(1)}^f, x_{max(2)}^f))$$
$$\mathbf{If}\ (y_{min(3)}^f > \text{minimum}\ (y_{min(1)}^f, y_{min(2)}^f)\ \&\&$$
$$(y_{max(3)}^f < \text{maximum}\ (y_{max(1)}^f, y_{max(2)}^f))$$
$$\mathbf{If}\ (z_{min(3)}^f > \text{minimum}\ (z_{min(1)}^f, z_{min(2)}^f)\ \&\&$$
$$(z_{max(3)}^f < \text{maximum}\ (z_{max(1)}^f, z_{max(2)}^f))$$

Two objects can have more than one static spatial relation regarding each other: e.g. one object's AABB can be both to the right and in front of other object's AABB. However, for forming the ESEC (as will be explained in III-D) we need only one relation per object pair. Here we propose a solution for this problem.

Fig.5. Defining betweenness by AABBs. In this scene, yellow AABB is between white and blue AABBs.

Each AABB is a cube with 6 rectangles. Let us label them as top, bottom, right, left, front and behind based on their positions in our scene coordinate system. Whenever object α_i is in the right of object α_j, one can make a projection from the left rectangle of object α_i onto the right rectangle of object α_j and consider only the rectangle intersection area which we will call "*shadow*" in this work.

Suppose $SR(\alpha_i^f, \alpha_j^f) = \{\gamma_1,\ldots,\gamma_k\}$ while $\{\gamma_1,\ldots,\gamma_k\} \in$ *Rel_static* and we have calculated *shadow* $(\alpha_i^f, \alpha_j^f, \gamma)$ for all relations γ between the objects α_i^f and α_j^f. The relation with the biggest *shadow* is chosen as the main static relation for the two objects:

$$SR(\alpha_i^f, \alpha_j^f) = \gamma_n, \mathbf{If}\ shadow(\alpha_i^f, \alpha_j^f, \gamma_n) = \max_{1\le m\le k}(Shadow(\alpha_i^f, \alpha_j^f, \gamma_m)).$$

The static relations around objects are highly dependent on the viewpoint and their changes, also do not make a human-notable difference in the performance of manipulation actions. For instance, when picking a knife to cut a cucumber we do not note if the knife is picked from the right or the left side of the cucumber. Thus we define a new relation called "Around" (**Ar**) and map the set of relations {**L**, **R**, **F**, **Ba**} onto it. In fact, "Ar" (Around) includes the space located on lateral sides of an object in a limited radius equal to threshold Ţ. This space does not cover the vertical neighborhood areas like "Above" or "Below".

TABLE 1: Definition of spatial relation directions

Directions	Right	Left	Front	Back	Above	Below
Relevant vector	+x	-x	+z	-z	-y	+y

Now we switch to explaining dynamic relations *DR* which we define as a two argument function where arguments are objects in the scene. When the distance between two objects' AABB decreases during a time segment (let us say within Θ frames; we have used Θ=10 in our experiments, given the 30 fps recording), they are "Getting Close" (**GC**) and when this distance increases, these two objects are "Moving Apart" (**MA**). Formal definition is given next, where the threshold τ is kept at 0.1 m:

$$DR(\alpha_i^f, \alpha_j^f)\ (i \neq j): \begin{cases} \mathbf{GC}, & \text{if: } \Omega(\alpha_i^{f+\Theta}, \alpha_j^{f+\Theta}) - \Omega(\alpha_i^f, \alpha_j^f) < \tau \\ \mathbf{MA}, & \text{if: } \Omega(\alpha_i^{f+\Theta}, \alpha_j^{f+\Theta}) - \Omega(\alpha_i^f, \alpha_j^f) > \tau \end{cases}$$

When calculating **GC** and **MA** we are also checking the touching relations $SR_{touch}(\alpha_i^f, \alpha_j^f) = (\mathbf{T}$ or $\mathbf{N})$ between the two objects. Based on SR_{touch}, we define two conditions required for calculating the remaining dynamic relations:

P1: $Rel\_touch(\alpha_i^f, \alpha_j^f) = \mathbf{T}$ **&&** $Rel\_touch(\alpha_i^{f+\Theta}, \alpha_j^{f+\Theta}) = \mathbf{T}$

P2: $Rel\_touch(\alpha_i^f, \alpha_j^f) = \mathbf{N}$ **&&** $Rel\_touch(\alpha_i^{f+\Theta}, \alpha_j^{f+\Theta}) = \mathbf{N}$

The third condition is on object α_i, α_j movement:

P3: $O_i^f \neq O_i^{f+\Theta}$ **&&** $O_j^f \neq O_j^{f+\Theta}$

The dynamic relations **MT**, **HT** and **S**, based on the three conditions above are defined in the following way:

$$DR\ (\alpha_i^f, \alpha_j^f)\ (i \neq j) \begin{cases} \textbf{MT,} \text{ if: } P1 \text{ and P3} \\ \textbf{HT,} \text{ if: } P1 \text{ and } \sim \text{P3} \\ \textbf{S,} \quad \text{if: } P2 \text{ and } \Omega\ (\alpha_i^{f+\Theta}, \alpha_j^{f+\Theta}) - \Omega\ (\alpha_i^f, \alpha_j^f) < \tau \end{cases}$$

D. Enriched SEC framework (ESEC)

As mentioned in the introduction, the Enriched SEC framework is inspired by the original Semantic Event chain (SEC) approach [4]. The original SECs check touching (T), not-touching (N) and absence (A) relations between each pair of objects in all frames of a manipulation scene and focus on transitions (change) in these relations. The extracted sequences of relational changes (represented in a form of a matrix, see first matrix in Fig. 6) are used in the manipulation action recognition. In the Enriched SEC framework the wealth of relations described in section III-C are embedded into a similar matrix-form representation, showing how the set of relations changes throughout the action. We expect to be able to differentiate actions in more details this way.

As the first step of making an Enriched SEC, we recognize so called "fundamental objects" among all of the other objects in a manipulation scene. Definition of these objects are based on the original SEC relations and given in Table 2. This way we exclude distractor objects which are present in the scene but do not perform any role in the manipulation.

TABLE 2. Definition of fundamental objects during manipulation action

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
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
Primary	The object from which the main object separates	Initially touches the main object. Changes its relation to not touching during the action
Secondary	The object to which the main object joins	Initially does not touch the main object. Changes its relation to touching during the action

As ESEC representation we introduce two matrices: one for representing the sequence of the static spatial relations *Rel_static* between the fundamental manipulated objects and one for describing the sequence of dynamic relations *Rel_dynamic* between the objects. We calculate static and dynamic relations in the sequence of the video frames of a manipulation action and add a new column to both (static and dynamic relation) matrixes whenever any static or dynamic relation has changed. This way we obtain a notation in matrix form as shown in Fig. 7 (middle is the static relation matrix and bottom is the dynamic relation matrix).

Alternatively, we can interpret our matrixes as sequences of graphs, where fundamental objects are connected by edges with the labels of static and dynamic relations. Each column in each matrix represents one graph, and the sequence of columns shows the time-development of those graphs.

One can observe (compare top representation in Fig. 6 for the original SEC with the bottom representation for the ESEC), that the ESEC has more columns as compared to the original SEC.

H, M
P, S
M, S

A N T T T T N N A
T T T T T T T T
N N N T N N N N N

H, M
P, S
M, S

A Ar Ab To To To To To Ab Ar A A
To To To To To To To To To To To To
Ar Ar Ar Ar Ar Ab To Ab Ar Ar Ar Ar

H, M
P, S
M, S

A GC GC GC MT MT MT MT MA MA MA A
HT HT HT HT HT HT HT HT HT HT HT HT
S S S S GC GC S MA MA MA S S

Fig.6. Description of a cutting action in SEC and Enriched SEC frameworks. First row: frames from the manipulation video for visualization of the action; second row: segmentation of the frames above, third row: SEC matrix, fourth row: ESEC: Static relation matrix; fifth row: ESEC: dynamic spatial relation matrix; knife is the main object (M), table is the primary object (P), cucumber is the secondary object (S).

E. Similarity measures and classification procedure

For calculating similarity "*sim*" between two ESECs we use a measure based on Longest Common Subsequence (LCS) as described in [8]. For discriminating different actions, we define, a threshold (Ɛ) according to the minimum similarity value between the ESEC matrices of the same manipulation actions in real data: $Ɛ = \text{Min}_k\ (\text{Min}_q\ (sim\ (A_{kq}, A_{kq})))$. Here A_k is a representative of a manipulation type (e.g. in the MANIAC data set we are using in further experiments) and A_{kq} indicates the q-th scenario of that action, in the dataset. For action classification we follow the online procedure defined in [8] in a slightly simplified way (see pseudocode in Fig. 7).

```
For (1 ≤ i ≤ 120)
   If (i=1)
      Make "cluster one" and assign ESEC_i to "cluster one"
   Else
      For (1 ≤ j ≤ Number of existing clusters)
         For (1≤ k ≤ Number of members in cluster j)
            S_ijk = sim (ESEC_i, m_jk) // calculate similarity to cluster j
member k
      S_i = max ({S_ijk})  // find maximum
      J = arg(max(S_ijk)) // find to which cluster maximum belongs
      If (S_i> Ɛ)
         Assign ESEC_i to cluster J
      Else
      Create new cluster and assign ESEC_i to the new cluster
```

Fig.7. Pseudocode for ESEC clustering.

We take ESECs extracted for each dataset video in a random order. The first ESEC is assigned to cluster one. For the second randomly selected ESEC we calculate the similarity *sim* to the first ESEC. If the similarity is above the threshold Ɛ, we assign the ESEC to the same cluster. Otherwise, we assign the ESEC to a new cluster. When more than one ESEC is already assigned to some cluster, we calculate the maximum similarity between the cluster members and the new ESEC. In case more than one cluster show above-threshold similarity, the ESEC is assigned to the cluster with the highest similarity. The procedure is continued until the dataset is exhausted. Afterwards class labels are assigned to clusters using the ground-truth labels, according to the majority in that cluster and the classification error is calculated in comparison to the ground-truth labels.

IV. EXPERIMENTS

A. Data Sets

Our action classification experiments were performed on the MANIAC dataset. It includes 8 different manipulation actions (Pushing, Hiding, Putting, Stirring, Cutting, Chopping, Taking, and Uncovering), each of which is presented in 15 different versions performed by 5 different human actors (overall 120 demonstrations). Actors were performing actions in different order, choosing from a set of 30 different objects and performing in differently configured scenes. Manipulation instances of each action have big variations in terms of manipulated objects, their poses, and followed trajectories.

To address a wider action variety, we have conducted additional experiments on a 26 actions set presented in [9]. Here, however, we did not have data recordings and thus were working on hand-made action models following the methodology suggested in [9].

B. Spatial relations accuracy

Here we begin with a brief evaluation of the performance of our spatial relation model. We asked three persons to indicate static spatial relations between pairs of objects from the set *Rel_static* on a set of 120 selected scenes in the MANIAC dataset. We have accepted the human-labeled relations to be the ground-truth in those cases where a majority vote was possible (2 matching human evaluations). We then calculated the relations using the algorithms introduced in III-C, including the extraction of the main relation, in case several relations were true, and compared with the ground truth. The obtained false positive rate is **FPR=4.725%** and the obtained false negative rate is **FNR=5.262%**.

C. Action Classification

We performed action classification on the MANIAC dataset as described in section III-E. The threshold Ɛ used for action discrimination for the MANIAC dataset is Ɛ=57%.

Table 3 compares action classification results of our novel ESEC representation to the results of the SEC framework as indicated in [8]. The classification accuracy for all actions is higher in ESECs. Totally, in average the spatial reasoning method has 97% accuracy in action classification which makes 12% improvement in compare of the previous method. This supports the notion that ESEC is a more powerful tool for classifying manipulation actions, as compared to the original SEC approach.

D. Discriminative ability of the Enriched SEC framework in an extensive actions set

A manipulation action ontology was designed in [9] where the hierarchical relations of 26 single-hand manipulation actions were based on the SEC framework (as well as pose and velocity considerations). However, it was shown that the discriminative ability of SECs alone is not enough to differentiate all those actions from each other. Here we will take the 26 manipulation actions analyzed in [26] and measure how much the discriminative ability increases when we use the ESECs for that purpose.

TABLE 3. Accuracy of classification on the MANIAC dataset in ESEC and SEC frameworks

Actions	ESEC	SEC[8]
Hiding	100%	87%
Pushing	94%	93%
Putting	100%	87%
Stirring	93%	93%
Cutting	91%	80%
Chopping	100%	93%
Taking	95%	87%
Uncovering	100%	80%
Average	**97%**	**85%**

The study [9] divides the 26 manipulation actions into six groups, where within one group all actions are similar or identical based on the SEC representation. Actions can be differentiated with SECs only *across* groups. Different from this, here we show how the Enriched SECs can now also differentiate actions *within* each group. Two groups are analyzed in Tables 4 and 5. To allow for fair comparison, we use as discrimination threshold 65% as this had been used in [8, 9]. As a consequence, in Tables 4 we see an action group where the ESECs differentiate the same number of actions as the SECs. However, ESECs can observe *sub-threshold differences* between the first three actions in the group, while the SECs indicate those actions as fully identical (similarity 100%).

In Table 5 we see an action group where ESECs can differentiate an additional action. Actions “cut” and “scoop” are 100% identical in the SEC representation, while the ESECs can differentiate those (with only **41%** similarity). We also see sub-threshold improvement when differentiating “Cut” from “Scissor cut”.

In addition, ESECs can differentiate actions “Put over” from “Push over” (**48%** similarity vs. 66% in SECs), “Break” from “Uncover by pick&place” (**18%** vs 69% in SECs), “Break” from “Uncover by pushing” (**19%** vs 69 in SECs), “Uncover by pick&place” from “Uncover by pushing” (**54%** vs. 67 in SECs).

V. DISCUSSION

In this paper, we have introduced a representation for manipulations and called the Enriched Semantic Event Chain, which focuses on spatial relations between objects in a scene. We divided possible spatial relations into “static” and

"dynamic" ones. For each action, the sequences of these static and dynamic spatial relations create a semantic descriptor of the manipulation action. The obtained descriptors are used to discriminate between different actions using real video sequences from the MANIAC data set (8 different actions) as well as sequences from the 26 actions from [9].

TABLE 4. ESECs showing differences in actions, when SECs indicate those as 100% similar (identical). "Hit&more" action set includes: Hit, Flick, Poke, Rub and Bore actions. Similarity values allowing action differentiation are shown in bold font.

ESEC					
Actions	**Hit&more**	**Push**	**Pull**	**Stir**	**Knead**
Hit&more	100%	83%	83%	**29%**	**46%**
Push		100%	83%	**19%**	**22%**
Pull			100%	**29%**	**46%**
Stir				100%	**0%**
Knead					100%

SEC					
Actions	**Hit&more**	**Push**	**Pull**	**Stir**	**Knead**
Hit&more	100%	100%	100%	**30%**	**60%**
Push		100%	100%	**30%**	**60%**
Pull			100%	**30%**	**60%**
Stir				100%	**44%**
Knead					100%

TABLE 5. ESECs differentiating between additional pair of actions, as compared to SECs. Similarity values allowing action differentiation are shown in bold font.

ESEC				
Actions	**Cut**	**Scissor cut**	**Draw**	**Scoop**
Cut	100%	83%	**52%**	**41%**
Scissor cut		100%	**14%**	**21%**
Draw			100%	**12%**
Scoop				100%

SEC				
Actions	**Cut**	**Scissor cut**	**Draw**	**Scoop**
Cut	100%	100%	**63%**	**100%**
Scissor cut		100%	**63%**	**42%**
Draw			100%	**36%**
Scoop				100%

Action differentiation by ESECs is compared to our earlier method based only on touching and not-touching events encoded in the older SEC (Semantic Event Chain) framework [8]. Both frameworks do not require object recognition and they ignore movement trajectories. Because in the original SECs touching and not-touching are the only defined spatial relations, the discriminative power of SECs is more limited than that of the here proposed Enriched SECs. This is shown by the difference between 96.625% action recognition accuracy for ESECs as compared to 87.5% for SECs using MANIAC. Also for the data from [9] we find improved performance and 5 more actions can be discriminated with ESECs. In addition, we found that several actions that had been 100% similar using the SEC framework begin to show differences when using ESECs (e.g. 83% similarity only). All this clearly shows that ESECs have a higher discriminative power than SECs. Because of this ESEC are necessarily also more robust against noise during action observation.

Evidently, there are some actions that can only be distinguished when considering dynamics, too (e.g. push versus hit). Those are not covered by the (E)SEC frameworks. In our older works [8,9] we had argued for a level-based semantic understanding of manipulations, where (E)SECs represent one certain symbolic level of understanding which can be supplemented by "finer" sub-symbolic layers (such as differentiating actions on the grounds of their different movement characteristics). ESECs help this process, because – having more transitions than SECs – they are breaking down an action into more (symbolic) components. Suppose we want to put a cup on the top of a box. In the original SEC, the relation between cup and box is initially "not-touching" and later "touching". With an ESEC representation there are additional phases where the cup is "Getting close" or is "Above", etc. These phases are now quite fine-grained and this should allow defining and joining trajectories for each phase. As the ESEC framework describes the sequence of required object relations based on quantitatively measured object (and manipulator) positions, it is possible to use the entries in the columns of the ESECs to provide quantitative start and end points for the manipulator trajectory. We had designed such a procedure using the older SEC framework coupled to DMPs [27] for trajectory generation in [28, 29] and we can now do the same in an improved way using the finer-grained representation of ESECs instead of the SECs.

Acknowledgements. 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.

REFERENCES

[1] Y. Yacoob, M. Black, "Parameterized modeling and recognition of activities," in *International Conference on Computer Vision*, 1998, pp.120-12.

[2] L. Lo Presti and M. La Cascia, "3D skeleton-based human action classification: A survey," *Pattern Recognition*, vol. 53, pp. 130–147, May 2016.

[3] B. Rosman and S. Ramamoorthy, "Learning spatial relationships between objects," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1328–1342, Sep. 2011.

[4] J. Goguen, "Mathematical models of cognitive space and time," in *Reasoning and Cognition: Proceedings of the Interdisciplinary Conference on Reasoning and Cognition*, 2006, pp. 125–128.

[5] M. Aiello and B. Ottens, "The Mathematical Morpho-logical View on Reasoning About Space," in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, San Francisco, CA, USA, 2007, pp. 205–211.

[6] K. Zampogiannis, Y. Yang, C. Fermüller, and Y. Aloimonos, "Learning the spatial semantics of manipulation actions through preposition grounding," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1389–1396.

[7] 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, Sep. 2011.

[8] 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, Sep. 2015.

[9] 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, Jun. 2013.

[10] T. M. Crockett, M. W. Powell, and K. S. Shams, "Spatial planning for robotics operations," in *2009 IEEE Aerospace conference*, 2009, pp. 1–7.

[11] J. A. Park, Y. S. Kim, and J. Y. Cho, "Visual reasoning as a critical attribute in design creativity," in *Proceedings of International Design Research Symposium*, 2006.

[12] M. Contero, F. Naya, P. Company, and J. L. Saorín, "Learning Support Tools for Developing Spatial Abilities in Engineering Design," *International Journal of Engineering Education*, vol. 22, no. 3, pp. 470–477, Jun. 2006.

[13] H. Schultheis, S. Bertel, and T. Barkowsky, "Modeling Mental Spatial Reasoning About Cardinal Directions," *Cognitive Science*, vol. 38, no. 8, pp. 1521–1561, Nov. 2014.

[14] S. Eagleson, F. Escobar, and I. Williamson, "Hierarchical spatial reasoning theory and GIS technology applied to the automated delineation of administrative boundaries," *Computers, Environment and Urban Systems*, vol. 26, no. 2–3, pp. 185–200, Mar. 2002.

[15] Y. Wei, E. Brunskill, T. Kollar, and N. Roy, "Where to go: Interpreting natural directions using global inference," in *IEEE International Conference on Robotics and Automation, 2009. ICRA '09*, 2009, pp. 3761–3767.

[16] B. Bredeweg and P. Struss, "Current topics in qualitative reasoning," *AI Magazine*, vol. 24, no. 4, p. 13, 2003.

[17] J. Renz and B. Nebel, "Qualitative Spatial Reasoning Using Constraint Calculi," in *Handbook of Spatial Logics*, M. Aiello, I. Pratt-Hartmann, and J. V. Benthem, Eds. Springer Netherlands, 2007, pp. 161–215.

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

[19] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from RGB-D videos," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 951–970, Jul. 2013.

[20] G. Gemignani, R. Capobianco, and D. Nardi, "Approaching Qualitative Spatial Reasoning About Distances and Directions in Robotics," in *AI*IA 2015 Advances in Artificial Intelligence*, M. Gavanelli, E. Lamma, and F. Riguzzi, Eds. Springer International Publishing, 2015, pp. 452–464.

[21] H. Kjellström, J. Romero, D. Martínez, and D. Kragić, "Simultaneous Visual Recognition of Manipulation Actions and Manipulated Objects," in *Computer Vision – ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman, Eds. Springer Berlin Heidelberg, 2008, pp. 336–349.

[22] K. Nagahama, K. Yamazaki, K. Okada, and M. Inaba, "Manipulation of multiple objects in close proximity based on visual hierarchical relationships," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 1303–1310.

[23] K. Ramirez-Amaro, E.-S. Kim, J. Kim, B.-T. Zhang, M. Beetz, and G. Cheng, "Enhancing human action recognition through spatio-temporal feature learning and semantic rules," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013, pp. 456–461.

[24] Y. Yang, C. Fermuller, and Y. Aloimonos, "A cognitive system for human manipulation action understanding," in *the Second Annual Conference on Advances in Cognitive Systems (ACS)*, 2013, vol. 2.

[25] D. R. Faria, R. Martins, J. Lobo, and J. Dias, "Extracting data from human manipulation of objects towards improving autonomous robotic grasping," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 396–410, Mar. 2012.

[26] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter, "Voxel Cloud Connectivity Segmentation Supervoxels for Point Clouds,"in *2013 IEEE conference on computer vision and pattern recognition,* 2013, pp. 2027–2034.

[27] T. Kulvicius, K. Ning, M. Tamosiunaite and F. Worgötter, "Joining Movement Sequences: Modified Dynamic Movement Primitives for Robotics Applications Exemplified on Handwriting," in *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 145-157, Feb. 2012.

[28] M J. Aein, E E. Aksoy, M. Tamosiunaite, J. Papon, A. Ude and Wörgötter, F. "Toward a library of manipulation actions based on Semantic Object-Action Relations," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2013, pp. 4555 - 4562 .

[29] M J. Aein, E E. Aksoy, F. Wörgötter, "Library of Actions: Implementing a Generic Robot Execution Framework by Using Manipulation Action Semantics", Submitted to *International Journal of Robotic Research (IJRR)*.

Chapter 3

Manipulation Actions Prediction Algorithm: Basic Idea and Implementation

In the previous chapter, we focused mainly on the representation and classification of manipulation actions using the ESEC framework.
This chapter includes another original paper highlighting another important application of the ESEC framework, *early recognition* or *prediction* of manipulations. Normally, automatic systems can only recognize actions only when it is finished, while here we provide an approach to predict actions, helping to yield to a system that provides the type of action well before an action has completed. This feature can be noticeably effective in interactions between humans and robots.
This paper includes the following:

- Categorization of manipulation actions according to their inherent ontology as well as their effects on the scene.
- Formulization of the prediction concept and its implementation on ESEC matrices.
- Definition of the prediction quantification measures for the analysis of theoretical and real data.
- Quantification against baseline method and a comparison between ESEC framework and other existing approaches for prediction of manipulations.
- Methods for human robot interaction to show that faster action prediction leads to a benefit in cooperation. To this end, two samples of human-robot interactions are chosen and implemented on a KUKA LWR robotic-arm.

2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

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 video sequences. 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 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.

∗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.

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] to complex actions in cluttered scenes or in realistic videos [6][7][8][9]. Also there are recent works in early event detection that have attempted to expand human action recognition towards action prediction [10][11][12][13][14]. These approaches try to predict actions from incomplete video data.

Ryoo [10] 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. [11] 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 [12] 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.

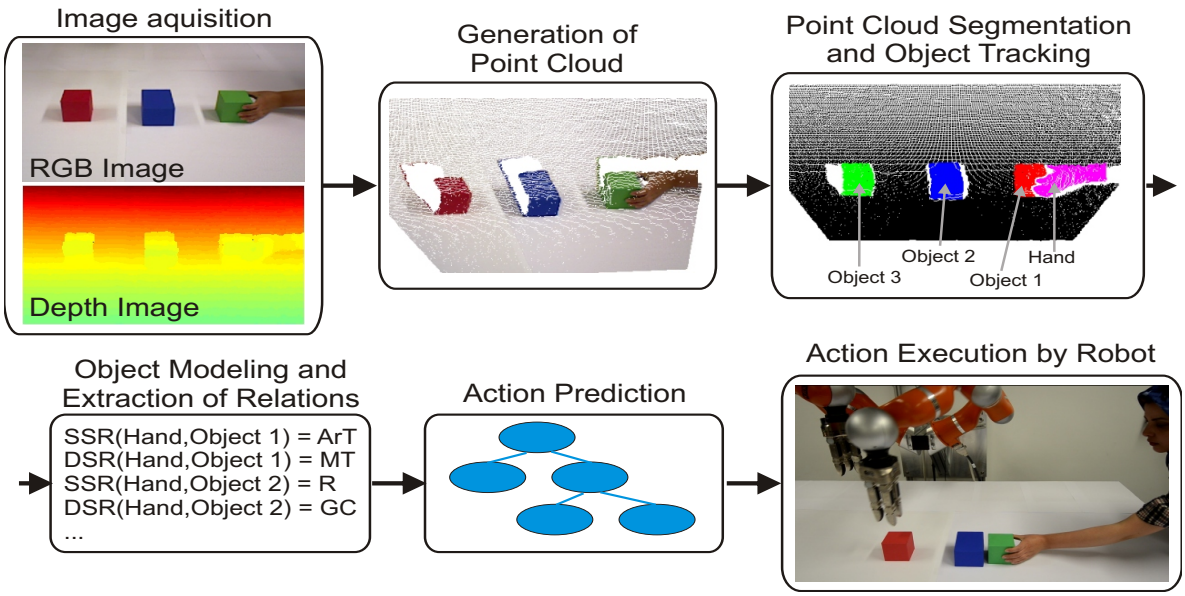


Fig. 1. Flow diagram of the prediction algorithm including human-robot interaction.

The importance of action prediction has been demonstrated recently in several robotic applications [13][14]. For example [13] 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. [14] 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 [10]. A more sophisticated state/action model is described in [11], who applied an adaptive Markov model to assign confidence regarding predictions of the human partners' actions.

Our focus in the current work is on visual prediction of manipulations, which are actions performed by hands. This is important for industrial as well as service robotics and also plays an essential role in human-robot interaction (HRI). Being able to efficiently and early predict, a robot will have more time to act accordingly and this way provide better adaptation in responding to human actions. Previous works mostly discuss about recognition of manipulations [3][15][16]. 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 the 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 [21] and [22]. 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 [20] [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.

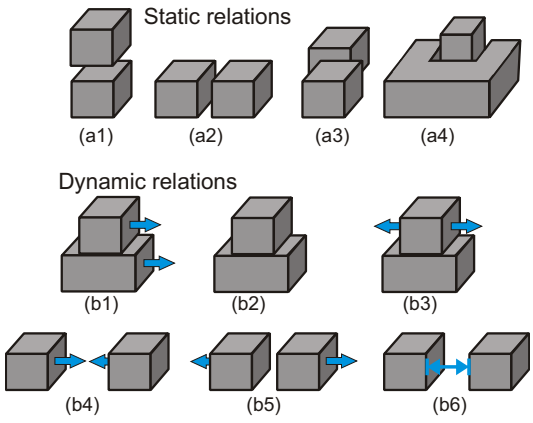
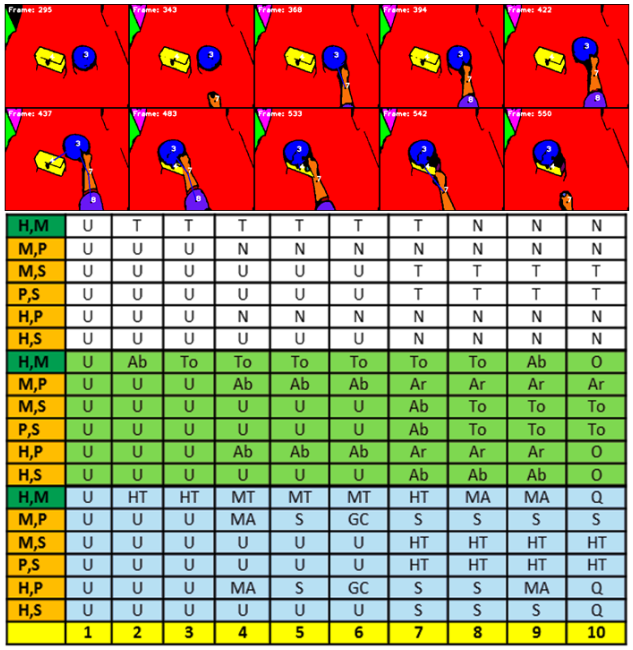


Fig. 2. (a) Static Spatial Relations: (a1) Above/Below, (a2) Right/Left, (a3) Front/Behind, (a4) Around. (b) Dynamic Spatial Relations: (b1) Moving Together, (b2) Halting Together, (b3) Fixed-Moving Together, (b4) Getting Close, (b5) Moving Apart, (b6) Stable.

[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. H: Hand, M: Main Object, P: Primary Object and S: Secondary Object, U: Undefined, T: Touch, N: Not-touch, Ab: Above, To: Top, Ar: Around, S: Stable, GC: Getting Close, MA: Moving Apart, MT: Moving Together, HT: Halting Together, O and Q: Not having a specific static and dynamic spatial relations, respectively. 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; yellow cells - show the number of events (when at least one static or dynamic spatial relation is changed in consecutive frames). The ESEC framework uses the whole table, while the SEC framework only includes the white part.

IV. Detailed Methods

A. Object Modeling

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: $\lambda_1^f, ..., \lambda_N^f$, where N is the number of objects and λ_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 [23].

Both static and dynamic spatial relations between two objects can be extracted simultaneously by evaluating the relations between AABBs of the objects as described in [3]. 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 frame 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 together (MT), are constant (HT), or one object (upper or lower)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}.

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.

C. Action Representation by ESEC

The ESEC framework is inspired by the original Semantic Event Chain (SEC) framework [18]. 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!). We have supposed that the hand only touches one object during a manipulation, therefore there is only one main object and the primary and secondary objects are considered unique, as well. 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).

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*

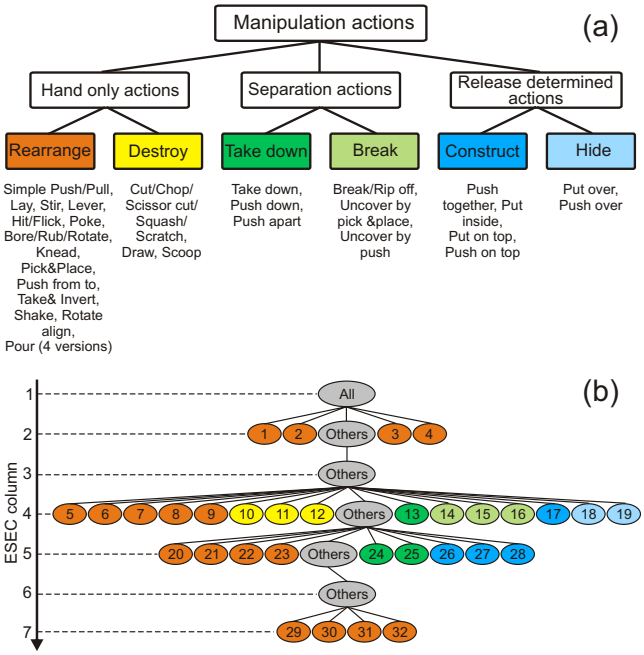


Fig. 4. (a) Categorization of 32 manipulation actions. (b) Prediction tree of manipulation actions according to ESEC framework.
Tree levels (1 to 7): show the ESECs column numbers that their corresponding actions become predictable in that column. All: mentions to the full list of manipulation actions. Others: mentions to the list of manipulations which are not yet distinguishable.

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 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.

Prediction occurs via comparison of the observed spatio-temporal relation sequence with the matrices in the action ontology according to the maximum similarity which is measured based on Longest Common Subsequence method (LCS)[20]. 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 [22] 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.

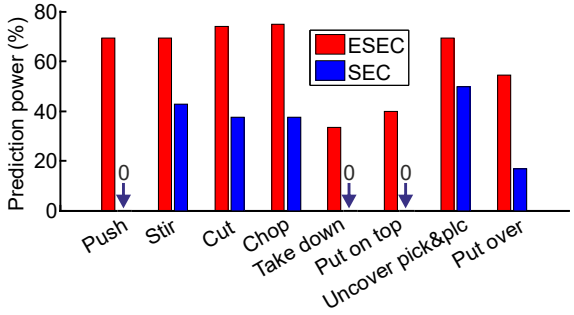


Fig. 5. SEC vs. ESEC in theoretical prediction analysis ($P_E(\alpha_i)$) on MANIAC data set actions.

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. 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.

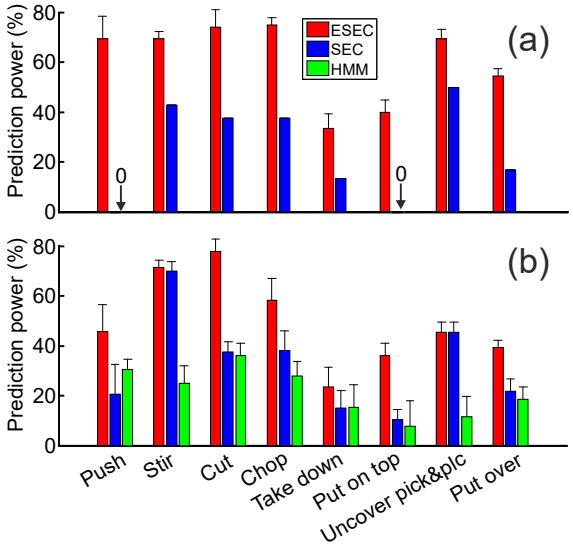


Fig. 6. Results of the comparison of action prediction using SEC and ESEC as well as the HMM methods on the MANIAC actions. (a) Event-based prediction ($P_E(\alpha_i)$). (b) Frame-based prediction ($P_F(\alpha_i)$). The error bars show standard deviations.

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.

2) Theoretical Analysis of MANIAC-Type Actions: A comparison of the theoretical prediction power between SEC

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%

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.

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 [20]. 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, the moments when an ESEC recognizes an action are earlier and they can, thus, predict faster than SECs and the HMM-based method. 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 average 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 [12][13][14] 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.

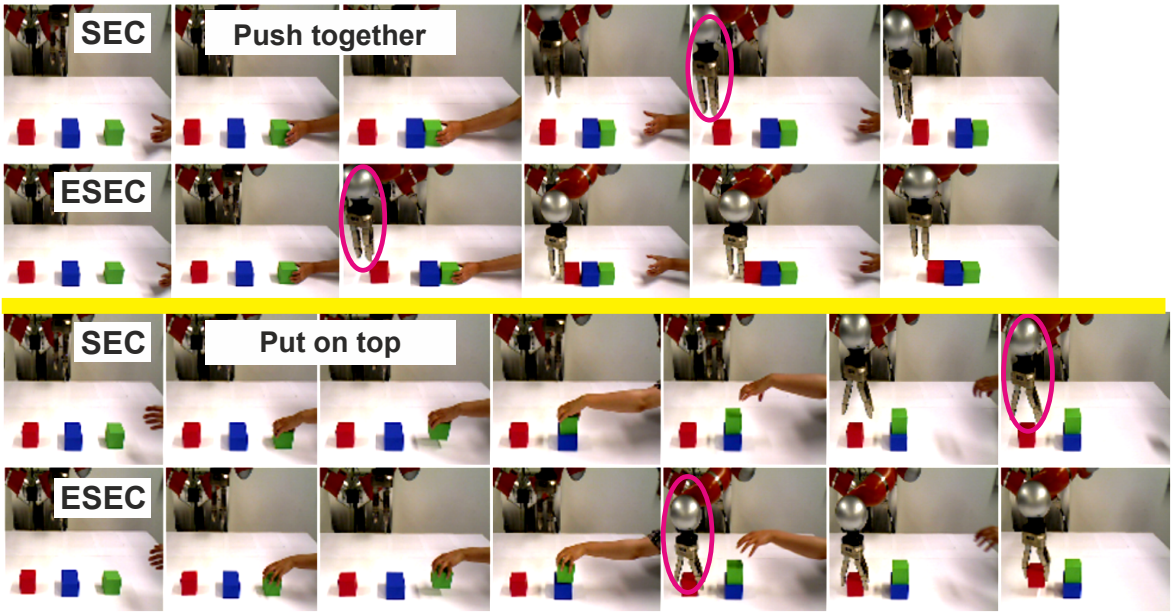


Fig. 7. Results from robot experiments. Ellipses mark stages at which the robot engages into an action. For more details please refer to the main text.

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.

[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. 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Y. Yang, C. Fermüller, and Y. Aloimonos, "Detection of manipulation action consequences (MAC)," in Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition. Portland,OR: IEEE, pp. 25632570, 2013.

[16] K. Zampogiannis, Y. Yang, C. Fermüller, and Y. Aloimonos, "Learning the spatial semantics of manipulation actions through preposition grounding," in 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 13891396, 2015.

[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] 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.

[20] 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.

[21] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter, "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.

Chapter 4

Recognition and Prediction of Manipulation Actions: Extended Idea, Complete Implementation and Comparison

The previous two chapters discussed the basis of our ESEC framework (object modeling, spatial reasoning, transition matrices, quantification measures and etc), as well as its application in classification and prediction of manipulation actions. We further compared ESEC with SEC and another hand trajectory pattern recognition algorithm and concluded our discussion with reference to the human-robot interaction experiment.
This chapter describes an enrichment of our previously introduced framework leading to a lower cost and higher accuracy and efficiency. We further compare the ESEC framework with other existing methods on real data .
This chapter includes an original manuscript consisting of the following:

- Noise detection and reduction algorithm for the ESEC manipulation action matrices according to a Probabilistic Context Free Grammar.
- Definition of the new fundamental object roles which leads to a higher accuracy in action representation, allowing us to differentiate between different actions, previously considered as identical.
- Definition of the new similarity measurement algorithm that significantly reduces the time and complexity of the calculations. This attribute considerably enhances the performance of manipulation action prediction.
- Comparison against a "Hidden Markov Model" (HMM) for hand motion recognition
  - Enhancement of hand trajectories with "Doglas Peucker" and "Dynamic Time Warping" (DTW) pre-processing algorithms.

  - Extension of the recognition concept to prediction.

- A comprehensive comparison between ESEC and SEC and the HMM based method on two publicly available point cloud manipulation action data-sets.

Robotics and Autonomous Systems 110 (2018) 173–188

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Recognition and prediction of manipulation actions using Enriched Semantic Event Chains

Fatemeh Ziaeetabar [a], Tomas Kulvicius [a], Minija Tamosiunaite [a,b], Florentin Wörgötter [a,*]

[a] Göttingen University, Institute for Physics 3 - Biophysics and Bernstein Center for Computational Neuroscience, Friedrich-Hund-Platz 1, 37077 Göttingen, Germany

[b] Vytautas Magnus University, Department of Informatics, Vileikos 8, 44158 Kaunas, Lithuania

HIGHLIGHTS

- We present a new algorithm for a prediction of manipulation action classes.
- Actions are represented by a matrix called Enriched Semantic Event Chain (ESEC).
- ESEC describes changing static and dynamic spatial relations between the objects.
- Actions can be correctly predicted after (on average) 45% of their execution time.
- Proposed approach outperforms a standard HMM-based method used for comparison.

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Action classification
Action prediction
Symbolic framework

ABSTRACT

Human activity understanding has attracted much attention in recent years, because it plays a key role in a wide range of applications such as human–computer interfaces, visual surveillance, video indexing, intelligent humanoids robots, ambient intelligence and more. Activity understanding strongly benefits from fast, predictive action recognition. Here we present a new prediction algorithm for manipulation action classes in natural scenes. Manipulations are first represented by their temporal sequence of changing static and dynamic spatial relations between the objects that take part in the manipulation. This creates a transition matrix, called "Enriched Semantic Event Chain (ESEC)". We use these ESECs to classify and predict a large set of manipulations. We find that manipulations can be correctly predicted after only (on average) 45% of their total execution time and that we are almost twice as fast as a standard HMM-based method used for comparison.

1. Introduction

Many methods have recently been developed for action recognition and many benchmark data sets have been prepared for measuring the performance of these methods [1–5]. Most of the computational approaches use segmented video as input and produce candidate action labels as output. These approaches usually consider action recognition as a classification issue.

The majority of the existing methods for human activity recognition focus on low-level spatio-temporal features, which can be brittle, for example due to problems of intra class variability arising from different humans performing the same action [6]. We, on the other hand, will not have problems recognizing actions performed by different people. Hence, evidently, humans are not troubled by the variability of low-level features present in movement trajectories, objects, and scene context. Approaches that use higher-level features [7,8] also seem to be less affected by this.

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

For prediction, variability [9] and incompleteness of the action execution [10] amplify the known problems in action recognition. After all, prediction is just "recognition earlier in time".

In this study we focus on visual recognition and prediction of manipulation actions, which are important for industrial as well as service robotics and also play an essential role in Human–Robot Interaction (HRI). To achieve this, we develop the so-called Enriched Semantic Event Chain (ESEC) framework [11], which is a much extended version of the Semantic Event Chain (SEC) [12]. ESECs use different static relations such as "around, above, below, inside", etc., and object movements like "getting close, moving apart", etc.,

* Corresponding author.
E-mail address: worgott@physik3.gwdg.de (F. Wörgötter).

https://doi.org/10.1016/j.robot.2018.10.005
0921-8890/

without specifying the fine details of object type, placement and motion. Hence, the framework remains *symbolic* and uses a representation, which also we might use when *speaking* about an action. Thus, ESECs are transition matrices, which symbolically encode the relational static and dynamic changes between (unspecified) objects.

The here presented framework allows comparing the development of the ESECs of different actions along the time-line, leading to a system that provides action-class 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.

After discussing the state of the art, in the following we will introduce and quantify the performance of the ESEC framework also in comparison to another prediction method that relies on the often-used Hidden Markov Model (HMM) approach. The symbolic character of ESECs allows in addition to define a Context Free Grammar for noise reduction further improving our approach.

2. Related works

In this section, we will review studies related to our work covering the following aspects: (a) Spatial Reasoning, (b) Human Activity Recognition and Prediction, (c) Semantic Representation and Recognition of Manipulation Actions, and (d) Prediction of Manipulation Actions.

(a) Spatial Reasoning: In this study we are specifically concerned with the analysis of relations between objects. Apart from the here investigated problem of manipulation understanding, this topic is also central to fields dealing with spatial representations and spatial reasoning (for example in: robot planning and navigation [13], interpreting visual inputs [14], computer aided design [15], cognitive science, geographic information systems (GIS) [16], natural language understanding [17], and several others). All of these cases need to represent and reason about spatial aspects of the world.

In robotics, one of the key aspects which is needed to understand commands such as "go in front of the closet door", is the ability to reason about spatial directions and relations in a quasi-human manner. In other words, the robot needs to be able to reason about an object with respect to another object in a given reference frame [18]. Therefore, finding spatial relations between objects in a scene is fundamental for the execution of tasks by robots.

Much of the above cited research also uses spatial relations in combination with a time-concept to structure spatio-temporal features, which can lead to semantic (relational) representation of the world to be used in the different applications. The next subsection shows that such (usually low-level) spatio-temporal features are indeed very helpful for addressing complex tasks.

(b) Human Activity Recognition and Prediction: One field which is strongly forced to fall back on spatio-temporal representations is human activity recognition and prediction. This could be simple human actions in constrained situations [19–22] up to complex actions in cluttered scenes or in realistic videos [23–26]. Also, there are recent works in early event detection that have attempted to expand human action recognition towards action prediction [27–31]. These approaches try to predict actions from incomplete video data.

Ryoo [27] proposed a method which explains each activity as an integral histogram of spatio-temporal features. Their recognition methodology, named dynamic bag-of-words, considers the sequential nature of human activities and uses those for prediction of ongoing activities.

Cao et al. [28] 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 re-creating the missing 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 [10] proposed a structured support vector machine (SVM) learning method to simultaneously consider both, local and global, temporal dynamics of human actions for action prediction. In another study [29] 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 importance of action prediction has been demonstrated recently in several robotic applications [30,31]. For example [30] anticipates future activities from RGB-D data by considering human–object interaction. This method has been tested in a real robot system employed 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. [31] used probabilistic modeling 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 for 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 [27]. This low-to-high level transfer may, however, often not really hold. A more sophisticated state/action model is described in [28], who applied an adaptive Markov model to assign confidence regarding predictions of the human partners' actions.

Most of the above cite work relies on rather fine-grained features. An alternative are feature sets, which are more strongly decoupled from the details of the scene. Many of the next-discussed studies use such features addressing the problem of manipulation understanding.

(c) Semantic Representation and Recognition of Manipulation Actions: Manipulation recognition can be understood as a sub-set within the above-discussed more general problem of human activity recognition. It has been addressed in different ways in several interesting studies [32–35].

In [32] functional object categories are extracted from spatiotemporal patterns, which encode interactions between hand and objects. The works in [33,34] try to explore a reasoning method, which extract semantic action rules by employing abstract hand movements with the object information and enhance manipulation actions recognition through spatio-temporal feature learning. In [36] visual semantic graphs are introduced for recognition of manipulation consequences according to the changes in the topological structure of the manipulated objects. The work in [37] modeled human manipulations by involving some semantic information about human skeleton and tracking the segments of manipulated objects and [38] used hand trajectories and hand–object interaction in a Bayesian model for manipulation understanding.

Aksoy et al. in [35] describe a method for semantic segmentation and recognition of long and complex manipulation actions, which captures the underlying spatiotemporal structure of an action and extracts basic primitive elements of each parsed manipulation [12]. Building on this, a more descriptive set of spatial relations was introduced in [11] (see also [39]).

(d) Prediction of Manipulation Actions: Our focus in the current work is not only to recognize but also to quickly predict manipulations. Recently Fermüller et al. have developed a recurrent neural network based method for manipulation action prediction [8]. They depicted the hand movements before and after contact with the objects during the preparation and execution of actions and applied a method based on a recurrent neural network (RNN) where patches around the hand were used as inputs to the network. They additionally used the estimations of forces on finger tips during the different manipulations for achieving more accurate predictions. Moreover, there are some studies about hand motion trajectory recognition, which work in a causal way and can be also used for prediction. For example [40,41] use a hidden Markov model-based continuous gesture recognition system utilizing hand motion trajectories. We have here extended their methods from recognition to prediction and compared it with our ESEC approach.

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 detailedness. Instead, we prefer using relational descriptions like "X moves toward Y", or "X is on top of Y". We may add "... moves 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 [12,42,43], which makes this system robust against individual spatial and temporal variations in the actual action execution.

3. Overview of the algorithm

Before explaining details of our method, first we provide an overview of the different steps of the algorithm (see Fig. 1).

First, all frames of a manipulation video are extracted. For each video frame, RGB and depth images from the Kinect device are used to generate 3D point clouds. These point clouds are then segmented and tracked by applying the algorithm presented in [44,45] according to color and depth information. First, all frames of a manipulation video are extracted. For each video frame, RGB and depth images from the Kinect device are used to generate 3D point clouds. These point clouds are then segmented and tracked by applying the algorithm presented in [44,45] according to color and depth information. The algorithm is called Voxel Cloud Connectivity Segmentation (VCCS) and is an over-segmentation algorithm for point clouds which uses voxel relationships and spatial connectivity to produce over-segmentation, which are fully consistent with the spatial geometry of the scene in three dimensional, rather than projective, space to help supervoxels conform better to object boundaries. Enforcing the constraint that segmented regions must have spatial connectivity, prevents label flow across semantic object boundaries, which might otherwise happen. Additionally, as the algorithm works directly in 3D space, observations from several calibrated RGB+D cameras can be segmented jointly. Thus, the VCCS algorithm uses region growing to produce uniformly sized supervoxels, while respecting object boundaries, inferred by large changes in local normals. The segments can then be tracked by warping the obtained segment labels to the next frame using real-time optical flow.

In addition to the point cloud data (used to determine physical object contact), we model each object using "Axis Aligned Bounding Box" (AABB) in order to assess spatial relations between objects (Section 4.4). Hence, no other information about object-type and/or its affordance is used in our recognition and prediction system. This allows us to deal with many scenes including various objects of different sizes, shapes, types and geometrical structures.

Next, we extract from the point-cloud data the information about which object is touching which other object. In addition, from the relative position and relative movement of these AABBs, static and dynamic spatial relations (SSR and DSR) are computed (Section 4.5). These are encoded as discrete entities (of which we have in total only 18), like "Above" or "Moving Together", etc. Hence, we do not consider continuous variables.

After that, we define the so-called Enriched Semantic Event Chain as an action descriptor (Section 4.2), which combines touching/non-touching information with the information about the spatial relations between all relevant object pairs (Section 4.3) in each movie frame. Only when any of these discrete relations *changes*, the corresponding event-change is stored as the next column in a transition matrix, the "ESEC" table. Hence, the ESEC table remains a very compact descriptor comprising not more then (about) 20 columns maximally.

The column-to-column transitions in an ESECs will always follow only certain rules (for example if an object is "above" another object then it cannot suddenly change to "below"). This allows us to define the Context-Free-Grammar (CFG) of ESEC-transitions. This is a very useful tool, because we can employ this CFG for noise-reduction. Evidently, using real data the computation of ESEC-relations is never 100% accurate due to noise in action execution as well as in the segmentation and tracking process. The ESEC-CFG allows immediately removing many evidently-nonsensical column transitions, which we do in the next step (Section 4.6).

As output we receive purified ESEC sequences, which can now be used for action recognition and action prediction. For this, we define a new method for similarity measurement between ESECs in Section 4.7 and this leads to our action clustering, classification and prediction methods, which are described in Sections 4.8–4.10. The prediction algorithm is a step by step procedure that utilizes the ESEC matrices in order to discriminate actions according to their event chains.

To demonstrate the quality of the ESEC-approach in comparison to others, in Section 4.11, we describe a standard baseline method for action classification based on Hidden Markov Model (HMM). This method is based on a hand gesture recognition procedure using two-level speed normalization, feature selection and classifier fusion based on [40,46] and extended to manipulation prediction by us.

4. Methods

4.1. Data sets

For experimental analysis, we used the MANIAC data set [42][1] and the KIT data set [47].[2]

The MANIAC data set consists of the following eight manipulation actions: *push, put, take, stir, cut, chop, hide (put over), and uncover*. Each action type is performed in 15 different versions by five human actors, resulting to 120 demonstrations. Each version has a differently configured scene with different objects and poses.

The KIT manipulation data set is a subset of the "KIT Whole-Body Motion Database" which has six action types: *cut, drink, mix, pick and place, pour, and put* with seven demonstrations per action type, resulting to 42 demonstrations.

For a theoretical analysis as well as for noise reduction procedures for real data we used an extended set of 35 manipulation

[1] http://www.dpi.physik.uni-goettingen.de/cns/index.php?page=maniac-dataset.

[2] https://motion-database.humanoids.kit.edu/.

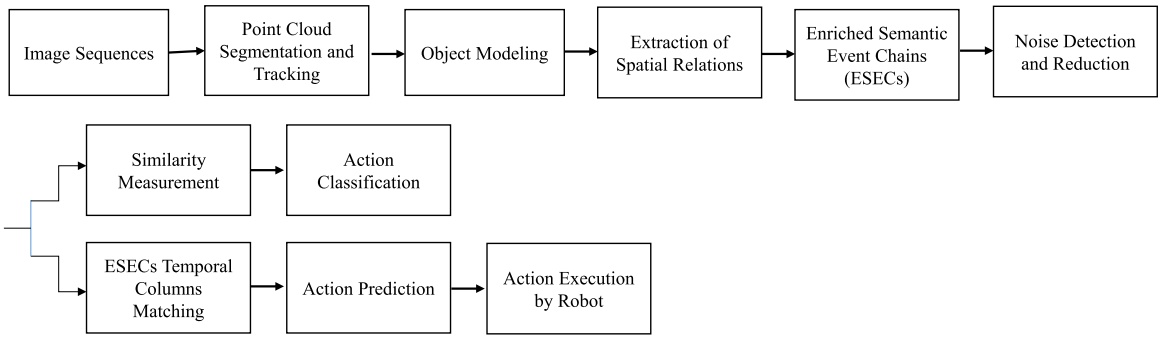


Fig. 1. Flow diagram of the algorithm.

actions, based on the smaller action set introduced in [43]. For the complete list of actions see caption of Fig. 9.

In [43], as well as in [8], it had been suggested that manipulations can be divided into three main groups (Fig. 2): "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 they can be subdivided into "Rearranging" (like push) and "Destroying" (like squash) actions. *Separation Actions* denote actions where the hand manipulates one object to remove it (or parts of it) from another object. This group is also divided into two cases: "Break" (e.g., ripping-off) and "Take-Down" (e.g., 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 is subdivided into "Hide" (e.g., covering an object with another one) and "Construct" (e.g., building a tower). According to this subdivision, here, we have analyzed and categorized 35 manipulation actions. For the theoretical analysis the event chains for all actions were manually created in an ideal and noise free way.

4.2. Enriched semantic event chain framework as an action descriptor

The core of our work relies on the Enriched Semantic Event Chain framework, the concept of which shall be introduced first before we describe all details of how to fill an ESEC matrix with events.

ESECs are inspired by the original semantic event chain (SEC) framework [12]. The original SECs investigated only the changes of touching (T) and non-touching (N) relations between all object pairs along a manipulation. A SEC is a matrix (table) where on the left side every row is indexed by the object pair to which this row refers and the core of the matrix describes the changes of the touching (T) and non-touching (N) relations for these object pairs over time. Hence, a new column is created whenever a change in N or T occurs and, as a consequence, every column reflects at least one such change. For example, the white upper rows of the matrix in Fig. 3 show the conventional SEC of an "uncovering" action.

These N–T-relational changes had been used in manipulation action recognition [12] but – as discussed later – this framework cannot recognize all the different 35 manipulations investigated here and it is also quite limited in its temporal-predictive power.

Here we still use these N–T-relations, too, but in the Enriched SEC framework we add a set of static (SSR) as well as dynamic spatial relations (DSR) in addition.

These spatial relations are shown by their abbreviations (see Section 4.5 for the definitions of all SSRs and DSRs), in a similar matrix-form representation in the lower two sections of Fig. 3. This figure, thus, shows how the set of all the different relations changes throughout an "uncovering" action.

Two aspects are needed to fully understand how an ESEC is generated: (1) What are the objects and their models? and (2) Which static and dynamic relations are used and how are they defined. This will be described next.

4.3. Object types

For this we introduce the concept of the so-called "fundamental objects", which are those that have an *essential role* in a manipulation action.[3] There are only five fundamental objects existing. Importantly, not all of them are always present in a manipulation. Their definitions are presented in Table 1. Also note that objects *obtain* their role through the course of the action. For example, 'fundamental object "2"': it is the *location in the sequence* of (N or T) transitions that lets some object become "2", which is that object that encounters the *second* transition.

This way, we naturally exclude irrelevant (distractor) objects in our manipulation and the ordering of the rows in an ESEC is always the same. Given five objects we have only $4 + 3 + 2 + 1$ possible relational combinations, resulting in ten rows for each of the subaspects (N/T, SSR, DSR) in the ESEC leading to thirty rows in total. Always, the upper ten rows denote N/T relational changes, while the middle and the bottom ten rows represent the sequences of SSR and DSR changes between each pair of fundamental objects in a manipulation, respectively. This ESEC matrix represents a detailed and precise action descriptor as demonstrated later for recognition and prediction of manipulation actions.

4.4. Object modeling as AABBs

For determining touching/non-touching we use the point-clouds. For definition all other spatial relations, a simpler object model suffices as defined next.

All coordinate axes are aligned according to the direction of the camera axes. The z axis corresponds to the depth direction (front/back), while x and y axes define direction of right/left and above/below, respectively. The camera is fixed during the manipulations and does not move. For simplicity, all relations have been defined relative to such a setting. Hence, if the camera moves one needs to transform the different relations. For example, if the view changes from a front-view to a back-view, relations left and right would invert, etc. All this, however, is straightforward and amounts to a reduced method for robotic coordinate system transformation. Using this definition, each object point cloud is approximated using an Axis Aligned Bounding Box (AABB). An AABB is a model that

[3] In our older works, we had still faced the complication that we needed to cover the N–T transition of all objects in a manipulation. This had led to a permutation problem, because objects had been arbitrarily labeled by the segmentation algorithm and movies with the same action performed twice could result in totally different label-order. Hence, to introduce *fundamental objects* is an important conceptual simplification.

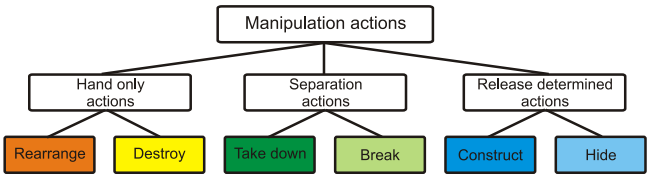
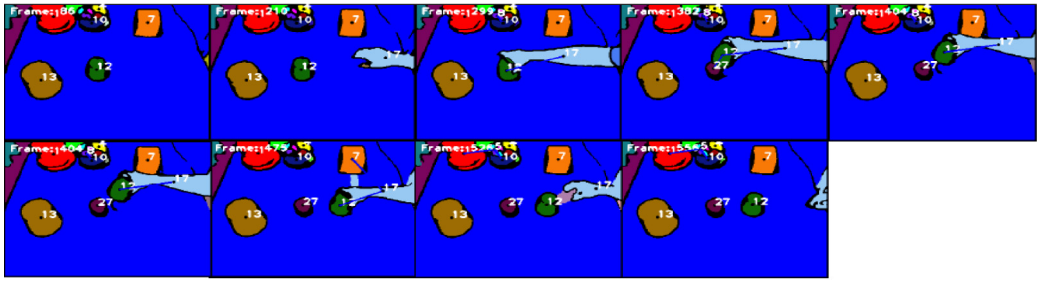


Fig. 2. Theoretical categorization of manipulation actions according to [43].



H, 1	U	U	T	T	T	T	T	T	N	N
H, 2	U	U	U	N	N	N	N	N	N	N
H, 3	U	U	U	U	U	U	U	U	U	U
H, G	N	N	N	N	N	N	N	N	N	N
1, 2	U	U	U	T	N	N	N	N	N	N
1, 3	U	U	U	U	U	U	U	U	U	U
1, G	U	U	T	N	N	N	N	T	T	T
2, 3	U	U	U	U	U	U	U	U	U	U
2, G	U	U	U	T	T	T	T	T	T	T
3, G	U	U	U	U	U	U	U	U	U	U
H, 1	U	U	To	To	To	To	To	To	Ab	O
H, 2	U	U	U	Ab	Ab	Ab	Ar	Ar	Ar	O
H, 3	U	U	U	U	U	U	U	U	U	U
H, G	O	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	O
1, 2	U	U	U	Ab	Ab	Ab	Ar	Ar	Ar	Ar
1, 3	U	U	U	U	U	U	U	U	U	U
1, G	U	U	To	Ab	Ab	Ab	Ab	Ab	Ab	Ab
2, 3	U	U	U	U	U	U	U	U	U	U
2, G	U	U	U	Ab	Ab	Ab	Ab	Ab	Ab	Ab
3, G	U	U	U	U	U	U	U	U	U	U
H, 1	U	U	HT	MT	MT	MT	MT	HT	MA	Q
H, 2	U	U	U	MA	MA	MA	S	S	MA	Q
H, 3	U	U	U	U	U	U	U	U	U	U
H, G	Q	S	S	MA	MA	S	GC	S	S	Q
1, 2	U	U	U	FMT	MA	MA	S	S	S	S
1, 3	U	U	U	U	U	U	U	U	U	U
1, G	U	U	HT	MA	MA	S	GC	HT	HT	HT
2, 3	U	U	U	U	U	U	U	U	U	U
2, G	U	U	U	HT	HT	HT	HT	HT	HT	HT
3, G	U	U	U	U	U	U	U	U	U	U

Fig. 3. Description of "Uncovering by pick and place" action in SEC and ESEC frameworks. Image frames (top): frame segmentation of an "Uncovering by pick and place" video. Green object (12) is object "1", table is the Ground (G) and purple object (27) is object "2". Event matrix (bottom): white cells are the SEC matrix; blue cells – ESEC static spatial relation matrix; green cells – ESEC dynamic spatial relation matrix. The ESEC framework uses the whole table, while the SEC framework only uses the white part.

circumscribes a point cloud by a cube with sides parallel to the directions of the coordinate system axes.

An example of a point cloud with its corresponding AABB is shown in Fig. 4 a. AABB computation details are discussed in [11].

4.5. Spatial relations

We have considered three types of spatial relations in this work: (1) "Touching" (**T**) and "Non-Touching" (**N**) relations, (2) "Static Spatial Relations" (**SSR**) and (3) "Dynamic Spatial Relation" (**DSR**).

T and N relations between two objects were determined based on occurrence (or non-occurrence) of a collision between the point-clouds [48], using kd-trees to speed up the evaluation [48].

SSR and DSR are extracted simultaneously by computing the relations between the AABBs of the objects.

Static Spatial Relations depend on the relative position of two objects in space. We do not need any data from previous frames for their evaluations and these relations are determined 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**), "Inside" (**In**), "Surround" (**Sa**) and "Between" (**Bw**). Right, Left, Front and Back are merged into "Around" (**AR**) and used at times when one object is surrounded by the other. 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. Fig. 5 (a1–a3) represents

Table 1
Definition of the fundamental objects during a manipulation action.

Object	Definition	Remarks
Hand	The object that performs an action.	Not touching anything at the beginning and at the end of the action. It touches at least one object during an action.
Ground	The object which supports all other objects except the hand in the scene.	It is extracted as a ground plane in a visual scene.
1	The object which is the **first** to obtain a change in its T/N relations.	Trivially, the first transition will always be a touch by the hand.
2	The object which is the **second** to obtain a change in its T/N relations.	Either T→N or N→T relational change can happen.
3	The object which is the **third** to obtain a change in its T/N relations.	Either T→N or N→T relational change can happen.

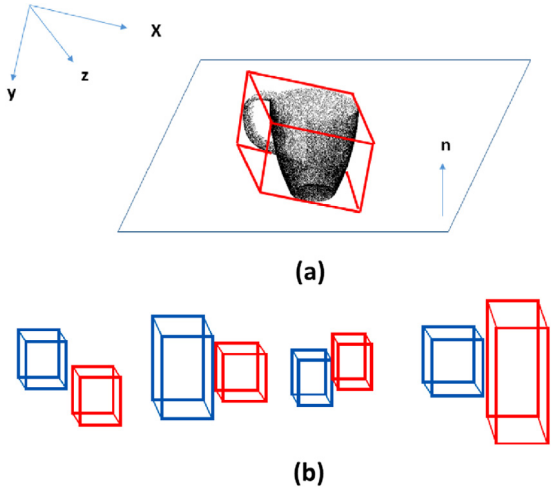


Fig. 4. (a) A point cloud and its corresponding AABB. (b) Possible states of Right–Left relations between two AABBs when size and y positions vary.

static spatial relations between two objects in terms of cubes. If two objects are far from each other or they do not have any of the above mentioned relations, their static relation is assumed as Null (**O**). This leads to a set of eleven static relations: **SSR** = {Ab, Be, R, L, F, Ba, Ar, Top, Bottom, ArT, In, Sa, Bw, O}.

Dynamic Spatial Relations define the spatial relation between two objects (moving in certain ways or not moving). 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 count for covering 0.5 s, 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 terms of cubes are shown in Fig. 5 (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 when 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

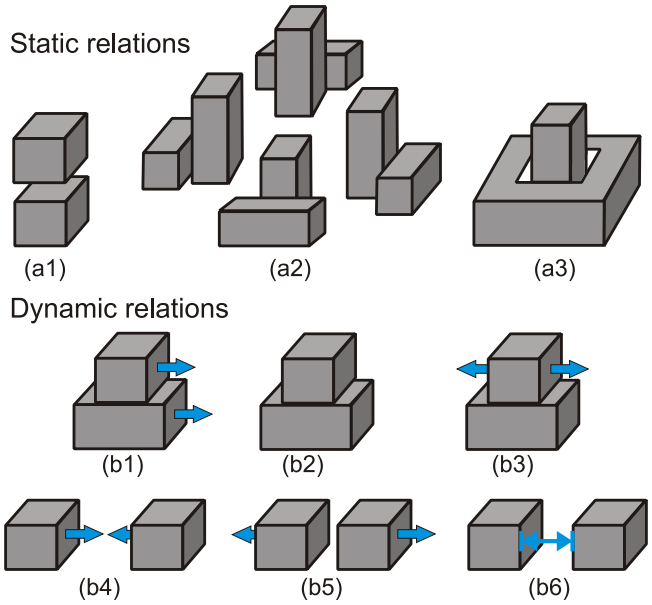


Fig. 5. (a) Static Spatial Relations: (a1) Above/Below, (a2) Around, (a3) Inside/Surround. (b) Dynamic Spatial Relations: (b1) Moving Together, (b2) Halting Together, (b3) Fixed-Moving Together, (b4) Getting Close, (b5) Moving Apart, (b6) Stable.

sequence. All these dynamic relations cases are clear from looking at Fig. 5 (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 do not have any of the above defined dynamic relations. Thus, we have a set of seven dynamic relations: **DSR** = {MT, HT, FMT, GC, MA, S, Q}.

Moreover, if one object becomes "Absent" or hidden during the actions (e.g. in put over, push over actions), we use **(A)** for annotating this condition. In addition, **(X)** is used if one object is destroyed or looses its primary shape (e.g. in cut, chop, scoop or break actions).

Each relation is defined by a set of rules. We start with specifying the rule set for static spatial relations. In general, $x_{min}, x_{max}, y_{min}, y_{max}, z_{min}$ and z_{max} are the minimum and maximum values between the points of the AABB of object α_i in x, y and z axes, respectively.

Let us consider the relation "Right": $SSR(\alpha_i, \alpha_j) = \mathbf{R}$ (object α_i is to the right of object α_j) if $x_{max}(\alpha_i) > x_{max}(\alpha_j)$ as well as all the following (exception) conditions are *not* true: $y_{min}(\alpha_i) > y_{max}(\alpha_j)$; $y_{min}(\alpha_j) > y_{max}(\alpha_i)$; $z_{min}(\alpha_i) > z_{max}(\alpha_j)$; $z_{min}(\alpha_j) > z_{max}(\alpha_i)$. The exception conditions exclude from the relation "Right" those cases when two objects-AABBs do not overlap in altitude (y direction) or front/back (z direction). Several examples of objects holding relation $SSR(red, blue) = \mathbf{R}$, when the size and shift in y direction varies, are shown in Fig. 4 b.

$SSR(\alpha_i, \alpha_j) = \mathbf{L}$ is defined by $x_{max}(\alpha_i) < x_{min}(\alpha_j)$ and the same set of exception conditions. The relations **Ab**, **Be**, **F**, **Ba** are defined in an analogous way. For **Ab** and **Be** the emphasis is on the "y" dimension, while for the **F**, **Ba** the emphasis is on the "z" dimension. For the relation "inside" $SSR(\alpha_i, \alpha_j) = \mathbf{In}$, x and z coordinates of AABB α_i must be between the x and z coordinates of AABB α_j respectively while $y_{min}(\alpha_j) < y_{max}(\alpha_i) \leq y_{max}(\alpha_j)$. The opposite holds for relation **Sa** (surrounding).

First we define the so called "Between Space" for two objects. This is obtained by extending the AABBs of two non-overlapping objects towards each other along our camera's axes and taking the intersection of those extensions. Whenever the third object's AABB completely stays in the "Between Space" of the two other objects', it is assumed that the third object is "in between" (Bw) of the two

objects. The rules for this relation in the case, are defined below $SSR(\alpha_i, \alpha_k, \alpha_j) = Bw$, (the object α_k is in between of objects α_i and α_j):

$x_{min}(\alpha_k) \geq minimum(x_{max}(\alpha_i), x_{max}(\alpha_j))$ and
$x_{max}(\alpha_k) \leq maximum(x_{min}(\alpha_i), x_{min}(\alpha_j))$ and
$y_{min}(\alpha_k) \geq maximum(y_{min}(\alpha_i), x_{min}(\alpha_j))$ and
$y_{max}(\alpha_k) \leq minimum(y_{max}(\alpha_i), x_{max}(\alpha_j))$ and
$z_{min}(\alpha_k) \geq maximum(z_{min}(\alpha_i), x_{min}(\alpha_j))$ and
$z_{max}(\alpha_k) \leq minimum(z_{max}(\alpha_i), x_{max}(\alpha_j))$

Two objects can have more than one static spatial relation regarding each other: e.g. one object's AABB can be both to the right and in front of the other object's AABB. However, for forming the ESEC we need only one relation per object pair. We solve this as follows.

Each AABB is a cube with six surfaces. Let us label them as top, bottom, right, left, front and back based on their positions in our scene coordinate system. Whenever object α_i is to the right of object α_j, one can make a projection from the left surface of object α_i onto the right rectangle of object α_j and consider only the rectangle intersection area, which we will call "shadow". Suppose $SSR(\alpha_i, \alpha_j) = \{Y_1, \ldots, Y_k\}$ while $Y_1, \ldots, Y_m \in SSR$ and we have calculated the $shadow(\alpha_i, \alpha_j, Y)$ for all relations Y between the objects α_i and α_j. The relation with the biggest shadow is then chosen as the main static relation for the two objects: $SSR(\alpha_i, \alpha_j) = Y_n (1 \leq n \leq k)$, $if: shadow(\alpha_i, \alpha_j, Y_n) = max_{1 \leq m \leq k}(Shadow(\alpha_i, \alpha_j, Y_m))$.

Static relations around objects are highly dependent on the viewpoint and the exact relation is often not relevant (also humans do not consider this many times). For instance, when picking up a knife to cut a cucumber we do not note whether the knife is picked up from the right or the left side of the cucumber. Thus, we define a different relation called "Around" (Ar) and map the set of relations L, R, F, Ba onto it. This way, "Ar" (Around) includes the space located lateral to the object in a limited radius equal to threshold ξ. This space does not cover vertical neighborhood areas like "Above" or "Below" [11].

Now we switch to explaining the dynamic spatial relations (DSR), which we define as a two argument function where arguments are the AABBs in the scene. Suppose Oi^f shows the central point of the AABB of object α_i^f (object α_i in f_{th} frame); we define $\delta(\alpha_i^f, \alpha_j^f) = \|Oi^f - Oj^f\|$ to be a two argument function for measuring the Euclidean distance between the AABBs α_i and α_j in f_{th} frame.

$$DSR(\alpha_i^f, \alpha_j^f) = \begin{cases} GC, & \text{if } \delta(\alpha_i^{f+\theta}, \alpha_j^{f+\theta}) - \delta(\alpha_i^f, \alpha_j^f) < \xi \\ MA, & \text{if } \delta(\alpha_i^{f+\theta}, \alpha_j^{f+\theta}) - \delta(\alpha_i^f, \alpha_j^f) > \xi \end{cases} \quad (1)$$

For this we use a time window of $\theta = 10$ frames in our experiments (recording speed is 30 fps); the threshold ξ is kept at 0.1 m:

When calculating **GC** and **MA**, we are also checking the touching relations between those two objects. For this we first define *TNR*, which is a two argument function which illustrates whether two objects are touching or non-touching. This is then used below to define several conditions:

$$\mathbf{P1}: TNR(\alpha_i^f, \alpha_j^f) = \mathbf{T} \&\& TNR(\alpha_i^{f+\theta}, \alpha_j^{f+\theta}) = \mathbf{T}$$
$$\mathbf{P2}: TNR(\alpha_i^f, \alpha_j^f) = \mathbf{N} \&\& TNR(\alpha_i^{f+\theta}, \alpha_j^{f+\theta}) = \mathbf{N}$$
$$\mathbf{P3}: O_i^f \neq Oi^{f+\theta}$$
$$\mathbf{P4}: O_j^f \neq Oj^{f+\theta}$$
$$\mathbf{P5}: \delta(\alpha_i^{f+\theta}, \alpha_j^{f+\theta}) - \delta(\alpha_i^f, \alpha_j^f) < \xi$$

The dynamic relations **MT**, **HT**, **FMT** and **S**, based on the three conditions above are now defined in the following way:

$$DSR(\alpha_i^f, \alpha_j^f) = \begin{cases} MT, & \text{if } P1 \&\& P3 \&\& P4 \\ HT, & \text{if } P1 \&\& \sim P3 \&\& \sim P4 \\ FMT, & \text{if } P1 \&\& (P3 \mathbf{XOR} P4) \\ S, & \text{if } P2 \&\& P5 \end{cases} \quad (2)$$

4.6. Noise detection and reduction

In theory each ESEC column describes one event as an essential part of a manipulation but in real data, when someone is carrying out a manipulation, there is variability and noise. To treat this, we use the fact that there are only certain column-to-column transition possible, while many others cannot exist (violating temporal causality or physics).

Hence, we define all possible transition rules for ESEC matrix-column transitions, based on hand-made noise-free ESECs of manipulation actions (see caption of Fig. 9 for the list of actions). The rules are given in the form of a Context-Free Grammar (CFG) and details of the CFG are described in Appendix A.

If a certain column-to-column transition in an ESEC does not satisfy any of the CFG rules it means an error has occurred. This allows detecting noise but we also need a method for noise reduction. To achieve this we modify the deterministic CFG into a *probabilistic* Context Free Grammar (PCFG) where each production rule is now assigned a certain probability (for definition of the PCFG see also Appendix A).

Once a noise-induced column has been detected, the best guess for a correction is to substitute the most probable item according to the transition probabilities as given in the PCFG (see Table 3 in the Appendix A). For example, assume we have the following transition in an ESEC matrix: $Ar \rightarrow Be$. We know that this is a wrong transition and according to Table 3 this transition is converted to: $Ar \rightarrow Ar$ as this is the most possible transition from Ar.

This method does not work perfectly and sometimes yields an incorrect transition (in cases where the correct transition corresponds to a rule which has not the highest probability), but all in all it still substantially reduces noise effects as shown in the results section.

4.7. Similarity measure

Next we discuss how to calculate the similarity of two manipulation actions. In the older SEC framework we had used the Longest Common Sub-sequence (LCS) method for similarity measurement [42], which we also have used in our previous work [11].

In this paper we define an improved similarity measure. This covers two new aspects: (1) different from before, here we now need to combine similarity assessments across three aspects: **N/T**, **SSR** and **DRS** transitions and (2) we are dealing with a rigorously temporally ordered set of (maximally) five fundamental objects: Hand, Ground, Object 1, Object 2 and Object 3. These objects have a strictly defined order of appearance, leading to a well-defined row ordering in the ESEC matrix.

Suppose θ_1 and θ_2 are the names of two actions with ESECs that have n and m columns, respectively.

Instead of writing down a 30-row ESEC each, we can concatenate the corresponding **T/N**, **SSR** and **DSR** of each fundamental object pair into a triple (f, g, h) and make a 10-row matrix for θ_1 and θ_2 with ternary elements instead. For θ_1 this reads (for θ_2 with elements $b_{i,j}$ accordingly):

$$\theta_1 = \begin{pmatrix} (a_{1,1}, a_{11,1}, a_{21,1}) & (a_{1,2}, a_{11,2}, a_{21,2}) & \cdots & (a_{1,n}, a_{11,n}, a_{21,n}) \\ (a_{2,1}, a_{12,1}, a_{22,1}) & (a_{2,2}, a_{12,2}, a_{22,2}) & \cdots & (a_{2,n}, a_{12,n}, a_{22,n}) \\ \vdots & \vdots & \ddots & \vdots \\ (a_{10,1}, a_{20,1}, a_{30,1}) & (a_{10,2}, a_{20,2}, a_{30,2}) & \cdots & (a_{10,n}, a_{20,n}, a_{30,n}) \end{pmatrix}$$

Using the elements of both matrices, we define the differences in the three different relation categories $L^{1:3}$ by:

$$L_{i,j}^1 = \begin{cases} 0, & \text{if } a_{i,j} = b_{i,j} \\ 1, & \text{otherwise} \end{cases}$$

$$L^2_{i,j} = \begin{cases} 0, & \text{if } a_{i+10,j} = b_{i+10,j} \\ 1, & \text{otherwise} \end{cases}$$

$$L^3_{i,j} = \begin{cases} 0, & \text{if } a_{i+20,j} = b_{i+20,j} \\ 1, & \text{otherwise} \end{cases}$$

where $1 \leq i \leq 10$, $1 \leq j \leq k$, $k = max(n, m)$

Then we define the compound difference for the three categories in the following way:

$$d_{i,j} = \sqrt{L^1_{i,j} + L^2_{i,j} + L^3_{i,j}}. \tag{3}$$

In case one matrix had more columns than the other matrix. i.e., $m < n$ or vice versa, we repeated the last column of the smaller matrix to match the number of columns of the bigger matrix.

Now we define D as the matrix, which holds all compound differences between the elements of the two ESECs.

$$D_{(10,k)} = \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,k} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ d_{10,1} & d_{10,2} & \cdots & d_{10,k} \end{pmatrix}$$

where $d_{i,j}$ denotes the dissimilarity of i_{th} objects pair at the j_{th} time stamp (column). Then, D, which is the total dissimilarity between ESECs of θ_1 and θ_2 is obtained as the average across all elements of matrix D.

$$D_{\theta_1,\theta_2} = \frac{1}{k * 10}\left(\sum_{j=1}^{k}\sum_{i=1}^{10} d_{i,j}\right) \tag{4}$$

Accordingly, the *similarity* between these ESECs Sim_{θ_1,θ_2}, is obtained as:

$$Sim_{\theta_1,\theta_2} = (1 - D_{\theta_1,\theta_2}) * 100\% \tag{5}$$

4.8. Action clustering

We performed automatic action clustering of 35 actions (see Section 4.1). This was done to assess how the *theoretical* categorization of actions based on their semantic meaning presented in Fig. 2 would match to the *automatic* action categorization (clustering) based on similarities between action tables for both ESEC and SEC frameworks.

For clustering, we employed the distance measure D_{θ_1,θ_2} between different actions (which is $1-Sim_{\theta_1,\theta_2}/100$). These distances D, calculated for SECs as well as ESECs, provide the input for a hierarchical clustering algorithm (also known as hierarchical cluster analysis [HCA]) were we used *complete linkage* (furthest distance) to build two hierarchical cluster trees, one for the ESEC- and the other for the SEC-framework. Here we used a distance threshold of 0.5 (corresponds to 50% similarity between actions) to break the tree into clusters. Thus, we group actions into one cluster if actions have more than 50% similarity (distance below 0.5).

4.9. Action classification

Here we are concerned with a action recognition. Action recognition is implemented by comparing the action table of a new action (test sample) to the action tables of existing action models (training samples) and computing the similarity score Sim_{θ_1,θ_2} as given above. We assign the class label to the tested action as the one belonging to the action, which had the maximal similarity score.

4.10. Action prediction and quantification measures

We have recently introduced the idea of manipulation actions prediction through spatial reasoning in [49]. We now decide to extend it in this paper.

Decision Tree: For the theoretical analysis of action prediction we build a decision tree from the manually defined event tables of the 35 manipulation actions (see caption of Fig. 9). This tree tells at what column of an ESEC an action can be unequivocally predicted. It is constructed in the following way: At the start, all first columns of the ESECs of the 35 manipulations are compared. Then, all actions with the same first column are categorized into the same set $(S_1, \ldots, S_n)$. Afterwards, the members of each set are compared according to their second column. 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 or all the columns of an action have been analyzed (see Fig. 9).

New Action Prediction: The same along-column comparison can now be done for any new action. In order to evaluate prediction performance we define a *prediction power* measure for the event based prediction in percent as:

$$P_E(\alpha_i) = \left(1 - \frac{E(\alpha_i)}{N_i}\right) * 100\%, \tag{6}$$

where $E(\alpha_i)$ is the *prediction event* column for an action $alpha_i$ at which the prediction of an action has actually occurred, and N_i is the number of columns in the matrix. 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 prediction at the start of the action.

Different from theoretically defined action tables, tables obtained from real data include noise and are not the same as theoretical ones. Therefore, in this case we perform prediction based on frames of action movies (frame based). For this, the **N/T**, **SSR**, and **DSR** relations are computed for each video frame. Similar to event based prediction, we perform column-wise comparison to action tables from the training data set (in this case we use several action tables as models for each action class) until all actions are categorized into a set which consists of the action members from the same class, or where there are no identical columns with any of the other actions. In the latter case, we compute the similarity measure as presented above for those incomplete action tables and predict the label based on the maximum similarity score. In case scores are identical for several action from different classes we proceed to the next column until a unique class is obtained.

Similar to above, the frame at which the prediction occurs is called *prediction frame* and is annotated as $F(\alpha_i)$. Accordingly, prediction power for the frame based prediction is defined as:

$$P_F(\alpha_i) = \left(1 - \frac{F(\alpha_i)}{L(\alpha_i)}\right) * 100\%, \tag{7}$$

where $L(\alpha_i)$, is the total number of frames during execution of an action α_i and denotes the duration of the action. The frames where the hand appears in the scene and leaves the scene are defined as the first and the last frame, respectively.

4.11. Comparison against baseline method

We compared our results with the performance of a state of the art HMM-based method from [40]. For a fair comparison we selected this method, because – like ours – it does not use object information, but, instead, relies on hand trajectories. To make the comparison even stronger, we improved the method from [40] by introducing noise reduction and feature fusion described in [46].

Table 2
Noise detection rate and correction rate among detected errors on MANIAC data set.

Actions	Noise detection rate	Noise correction rate
Put on top	56%	72%
Take down	59%	75%
Push	32%	84%
Cut	75%	66%
Chop	62%	53%
Stir	57%	49%
Put over	33%	61%
Uncover	48%	53%
Average	**52.75%**	**64.12%**

As this HMM-based method works in a causal way along the time line, instead of giving a result only at the very end, here this method has been used for prediction.

We have implemented this HMM-framework analyzing orientation, velocity, and location of the hand individually but also by fusing these results using majority voting. In addition, noise reduction and feature fusion [46] allowed removing noise such as hand trembling and unintentional movements to improve on the baseline.

As this is a very technical aspect and not related to our own methods, we are giving the details of the HMM implementation in Appendix B.

5. Results

In this section we present experimental results of our method and compare to SEC and HMM with respect to several different aspects. First, we measure the effect of the noise detection and reduction algorithm in the ESEC framework (Section 5.1). Second, the results on action clustering are presented for both ESEC and SEC frameworks in Section 5.2. After that, the results of manipulation action classification on the MANIAC data set are presented and compared to the SEC framework in Section 5.3. Finally, we have compared the performance of action prediction using ESEC against SEC and HMM-based method on MANIAC and KIT data sets (Section 5.4).

5.1. Noise reduction

Here we evaluated performance of the noise reduction technique proposed in Section 4.6 on the MANIAC data set. For that, we used all 120 actions from eight manipulation classes of the MANIAC data set, and compared automatically extracted ESEC tables from action videos to the theoretical (manually defined) ground-truth ESEC tables of the corresponding eight actions. By comparing the resulting matrices of the real with the ground-truth tables we identified *all* errors. In total there were **749** false events (errors). After that we calculated the percentage of the errors that we could actually detect using the CFG-method and also the percentage of errors that we were able to correct among those detected errors.

Table 2 presents the rates of noise detection and correction using our proposed CFG-based method. Results show that on average we were able to detect errors in **52.75%** of the cases, and correct **64.12%** of the detected errors. The False Positive Rate is **FPR=0%** as the noise correction algorithm only operates in case of not-allowed grammatical transitions, whereas the False Negative Rate is **FNR=47.25%**.

5.2. Clustering of manipulation actions

Unsupervised clustering of the semantic distances between the different actions is shown in Fig. 6 for SECs (right) and ESECs (left). Similarity between any two actions is encoded in these trees by the "height" on the distance axis that you have to overcome when climbing from one action to the other. Short vertical bars in the SEC diagram connect actions with zero semantic distance between them. These are actions that cannot be distinguished using SECs. This case does not exist in the ESEC diagram. The similarity tree for SECs essentially reproduces the results from our older study [43]. The red group on top contains the *Release Determined* plus *Take Down* actions (see Fig. 2 above). This corresponds well to our earlier findings of high intrinsic but also across-group similarities for these cases. The small purple group represents the *Break* actions and all remaining groups except the green one at the bottom are *Hand Only* actions, where light blue covers the *Destroy* group, dark blue the *Rearrange* group and orange is a mix from both. The green group contains all *Pouring* actions, which had not been considered in our older works. When comparing this tree to the confusion matrix shown in Fig. 9 in [43] one can see that, with few exceptions, there is a very high match. The "outliers" are those cases that were also outliers in our old confusion matrix.

In general, ESECs (left) discriminate more strongly while preserving the general picture very well, but now – as an important additional point – we observe that all actions can be separated from each other. There are no cases with zero distance anymore. For example, the red group (*Release Determined* actions) is basically preserved but now the *Take Down* subgroup (marked with a bracket) forms a clearly visible sub-cluster. It is important to realize that the here shown colored clusters are created when using a cutting threshold of distance=0.5. This corresponds to 50% similarity, a value, which we had used also in many other studies. When considering the ESEC tree, another "naturally looking" choice would be a threshold of 0.3. In this case the red and orange clusters would remain and all others would be split resulting in different semantic grouping.

In summary, these tree diagrams show that, depending on the threshold, different semantic groupings are observed. We find this interesting, because this reflects algorithmically also our own, human, way of considering action similarities: depending on intention, task, etc., we can also choose different "cutting thresholds" for what is similar and what is not. Also for us, action similarity semantics are not set in stone.

5.3. Classification of manipulation actions

We have performed action classification and compared classification accuracy of ESEC and SEC frameworks on the MANIAC data set (8 action classes) as described above. We performed Monte Carlo cross validation 20 times, where each time we randomly selected 10 different actions from each class (in total 80 actions) for training and used them as action models for comparison and 5 actions from each class (in total 40 actions) for testing. Confusion matrices are given in Fig. 7.

Bar plots for a comparison of classification performance between ESECs and SECs are shown in Fig. 8, where we show results for both, noisy event tables and noise-reduced event tables (i.e., by applying our noise reduction algorithm as described in Section 4.6 before classification). Results show that on the noisy data on average across all 8 action we obtained slightly better classification accuracy with ESECs than with SECs, **86.0%** (SD = 15.01%) and **76.75%** (SD = 19.54%), respectively. In this case, ESEC outperforms SEC in a recognition of three ("Take down", "Cut" and "Chop") out of eight actions (see Fig. 8). We obtained, however, a much better classification result by applying noise reduction where now (on average) classification accuracy was improved by **5.62%** for ESECs with a final score of **91.62%** (SD = 12.78%) and by **1.5**% for SECs with a final score **78.25%** (SD = 17.15%). In this case, classification accuracy for ESEC was significantly better in all, except the "Stir" action.

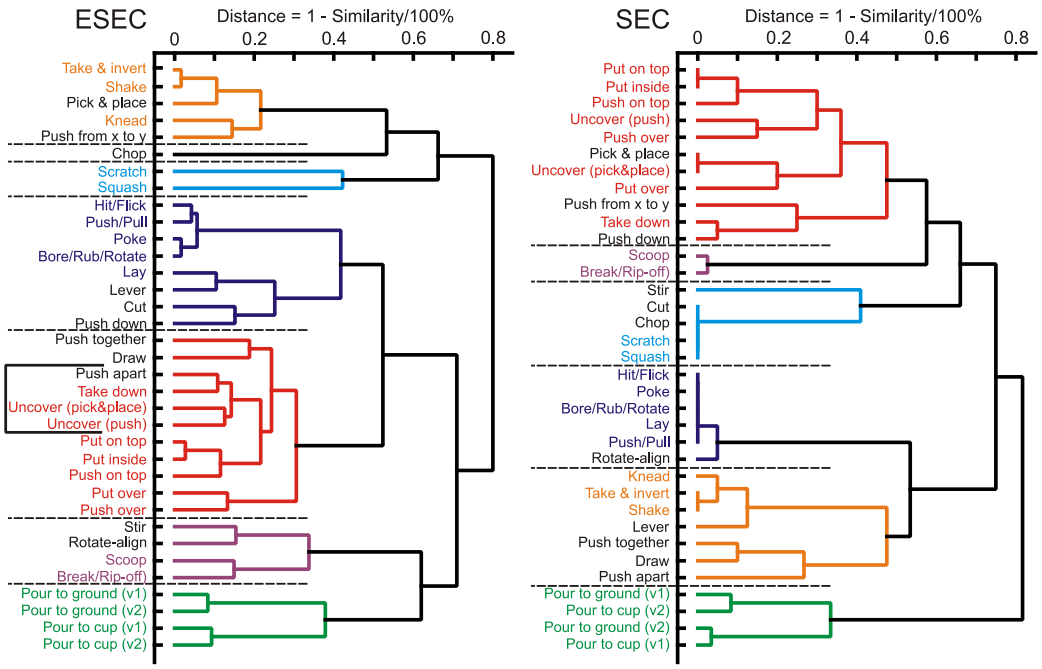


Fig. 6. Dendrogram of the hierarchical clustering of 35 theoretical actions based on ESEC and SEC frameworks. Here we used a distance threshold of 0.5 to cluster the actions. Note that colors mark clusters with the same action subsets.

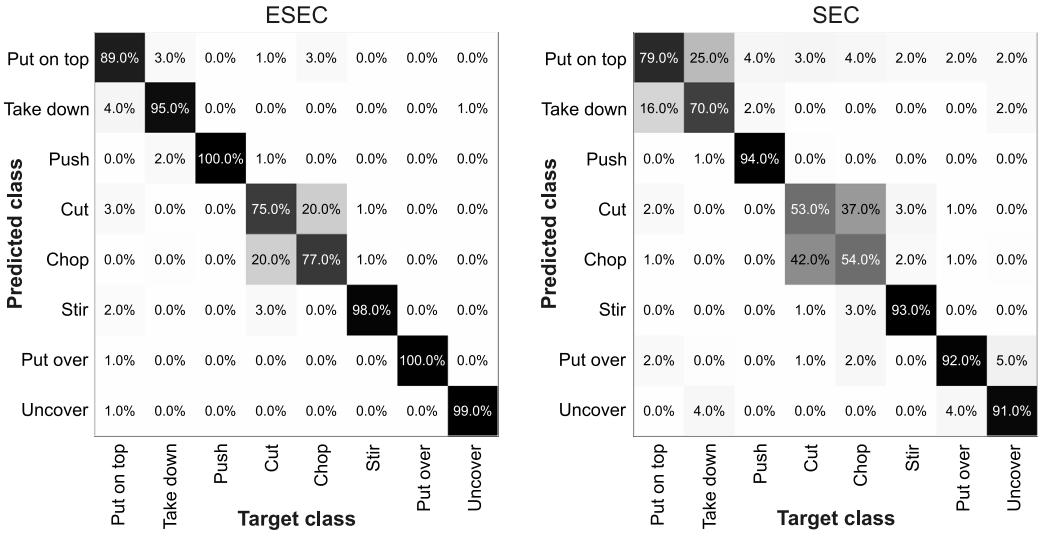


Fig. 7. Confusion matrices of classification (on-diagonal) and misclassification (off-diagonal) results on the MANIAC data set for ESEC and SEC frameworks for noise-reduced data. Average classification rate from 20 classification trials is shown for each target and predicted class pair.

The biggest recognition difference between ESECs and SECs is observed for "Take down", "Cut" and "Chop" actions. This is due to the fact that in case of the "Take down" action, in SEC this action is mixed up with the "Put on top" action, resulting in high misclassification rates of **25.0%** and of **16.0%**, see confusion matrices in Fig. 7, whereas in ESEC these errors are much lower, i.e., **4.0%** and **3.0%**. In case of the "Cut" and "Chop" actions, misclassification rates between these two actions for the SEC framework is even bigger (**42.0%** and **37.0%**). Note, in the theoretical SEC-action tables of these actions are identical! For the ESEC framework, action tables are more different, which results in lower misclassification rates (**20.0%** in both cases) and a better recognition of these two actions.

In summary, classification results demonstrate that on average ESECs with the noise reduction algorithm clearly outperformed SECs leading to a final improvement of **13.37%** in classification accuracy.

5.4. Prediction of manipulation actions

5.4.1. Theoretical analysis of all actions

For this analysis, we used the manually generated 35 ideal ESEC matrices for the representation of the manipulation actions.

First we show how action prediction evolves over time. For this we use the decision tree as defined in Section 4.10, shown in Fig. 9. This tree uses the same color code as in Fig. 2 and shows that maximally eight columns in an ESEC are needed until all actions are recognized. Note, the most complex action ("pouring") has in total 14 columns. Columns 1, 2, 6 and 7 have no added discriminative value. Four actions are found already in column 3, where 18 and

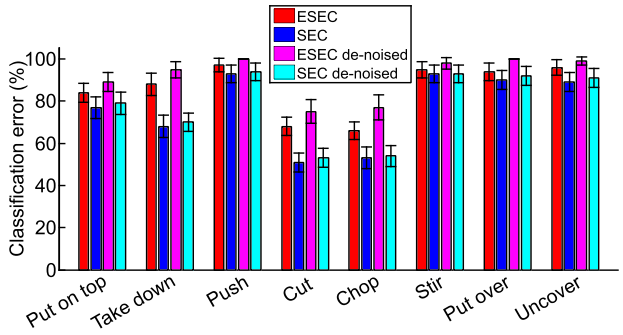


Fig. 8. Comparison of classification results on MANIAC data set between ESEC and SEC frameworks for noisy and noise-reduced (de-noised) event tables. Average classification accuracy and confidence intervals of mean (95%) from 20 classification trials are shown for each action class.

9 actions are discriminated in columns 4 and 5, respectively. The remaining four actions are recognized in column 8.

To quantify this better, we computed the event-based prediction power $P_E(\alpha_i)$ according to Eq. (6) for all 35 actions for both SEC and ESEC. We obtained an average prediction power of **43.94%** and **30.94%** when using the ESEC and the SEC, respectively (95% confidence interval of mean difference is [0.327 17.673]). This shows that we can predict actions using ESECs earlier (before half of the action has been completed) than when using SECs. Moreover, we could correctly predict all the 35 actions with ESECs (100.00%), whereas with SECs we could only predict 25 out of 35 actions (71.43%) correctly.

5.4.2. Prediction on real world datasets

Here we performed action prediction on two real world data sets: the MANIAC data set, the same on which classification was performed, and in addition the KIT data set, an independent data set, which had never been considered in any stage of the ESEC development before.

We compared prediction power of ESEC against prediction power of SEC and the baseline HMM-based method as described above. For the MANIAC dataset (8 action classes), as in the classification experiments, we performed Monte Carlo cross validation 20 times were each time we randomly selected 10 actions from each class for training (in total 80 actions) and 5 actions from each class for testing (in total 40 actions). For KIT data set (6 action classes), we also performed Monte Carlo cross validation 20 times were we used 4 randomly selected actions from each class for training (in total 24 actions) and 3 randomly selected actions from each class for testing (in total 18 actions).

Prediction on MANIAC dataset: Fig. 10 (top) shows the frame-based predictive power for the ESEC method compared to the SEC and the HMM methods for the eight MANIAC data set actions. The resulting average predictive power for the ESEC is **62.69%** (SD = 13.22%) out of **92.8%** correctly predicted actions, for the SEC it is **32.11%** (SD = 15.77%) out of **80.8%** correctly predicted actions, and for the HMM it is **34.48%** (SD = 10.56%) out of **70.2%** correctly predicted actions. Thus, ESECs have the highest predictive power on average and predict all eight actions earlier as compared to SEC and HMM. SEC is faster than HMM in predicting two ("Take down" and "Uncover") out of six actions, but slower than HMM in predicting "Put on top" and "Push" actions.

Prediction on KIT dataset: Fig. 10 (bottom) shows a comparison of all three methods (ESEC, SEC, and HMM) for the six KIT actions. In this case, average predictive power for the ESEC is **61.2%** (SD = 10.40%) out of **99.0%** correctly predicted actions, for SEC it is **39.82%** (SD = 11.5%) out of **82.33%** correctly predicted actions, and for the HMM it is **30.32%** (SD = 9.29%) out of **90.33%** correctly predicted actions. As for MANIAC, ESEC also here outperforms SEC and HMM methods in prediction of all six action classes. SEC is faster than HMM in predicting three actions ("Cut","Drink", and "Pick&Place").

In summary, prediction results demonstrate that on average our presented ESEC framework leads to earlier and more accurate predictions as compared to SEC and HMM methods when tested on two different data sets (in total 13 different actions), whereas SEC and HMM methods on average show similar prediction performance.

6. Discussion

In the following we will discuss the results of our framework and compare those to other frameworks.

6.1. Action discrimination

First, we compared our proposed method to the original SECs with respect to action discrimination in clustering and classification tasks. For clustering, we used 35 theoretical actions extended from [43] and for classification we used the MANIAC data set with eight different actions [42].

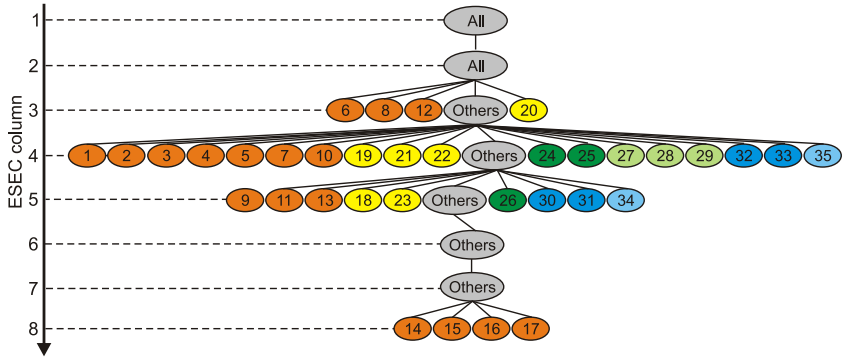


Fig. 9. Prediction tree of manipulation actions using ESEC framework. Colors correspond to action categories as shown in Fig. 2. List of actions:
(1) Hit/Flick; (2) Poke; (3) Bore/Rub/Rotate; (4) Lay; (5) Push/Pull; (6) Stir; (7) Knead; (8) Lever; (9) Push from x to y; (10) Take & invert; (11) Shake; (12) Rotate-align; (13) Pick & place; (14) Pour from a container onto the ground when the liquid first un-touches the container then touches the ground (Pour to ground [v1]); (15) Pour from a container on the ground when the liquid can touch the container and the ground at the same time (Pour to ground [v2]); (16) Pour from a container to another container when the liquid first un-touches the container then touches another container (Pour to cup [v1]); (17) Pour from a container to another container when the liquid can touch the container and another container at the same time (Pour to cup [v2]); (18) Cut; (19) Chop; (20) Scratch; (21) Squash; (22) Draw; (23) Scoop; (24) Take down; (25) Push down; (26) Push apart; (27) Break/Rip-off; (28) Uncover by pick & place; (29) Uncover by push; (30) Put on top; (31) Put inside; (32) Push on top; (33) Push together; (34) Put over; (35) Push over.

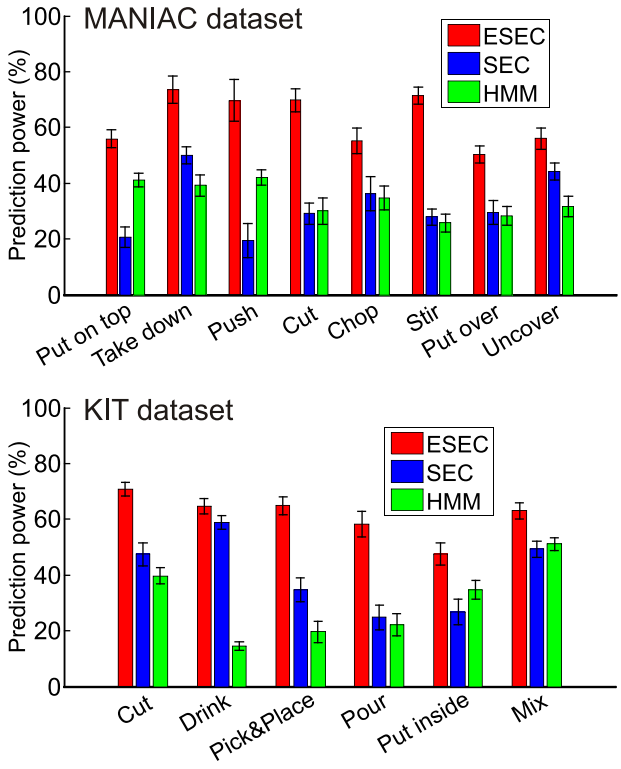


Fig. 10. Comparison of prediction results on the MANIAC data set (top panel) and the KIT data set (bottom panel) between ESEC, SEC, and HMM frameworks. Average frame-based prediction power and confidence intervals of mean (95%) from 20 classification trials are shown for each action class within correctly predicted actions.

When analyzing action clustering we found that ESECs discriminate actions more strongly than SECs. On the 35-action set, we obtained an improvement of **28.6%** where 10 more actions can be discriminated with ESECs (35 out of 35 actions) compared to SECs (25 out of 35 actions).

For action classification on average we obtained slightly better accuracy when using ESECs as compared to SECs: **86.0%** (SD = 15.01%) for ESEC vs. **76.75%** (SD = 19.5%) for SEC, where the improvement of the classification accuracy by ESECs was statistically significant for three out of eight MANIAC data set actions. This is due to the fact that, since in the original SECs touching and non-touching are the only defined spatial relations, the discriminative power of SECs is more limited than that of the here proposed ESECs. For example, when putting a cup on a box, in the original SEC, the relation between cup and box is initially "non-touching" and later "touching". In an ESEC representation there are additional phases, where the cup is "above" or "getting close" to the box, etc.

We then further improved these results by introducing grammar-based noise reduction techniques where we obtained substantially better classification accuracy for ESECs: **91.62%** (SD = 12.78%) for ESECs vs. **78.25%** (SD = 17.15%) for SECs. Here improvement in classification accuracy for ESEC vs. SEC is statistically significant for seven out of eight MANIAC data set actions. The noise reduction brought more improvement for ESEC as compared to SEC, because ESECs have more and more variable relations, thus they are more prone to errors. However, as demonstrated this can be nicely handled by the error reduction techniques as introduced in this study.

Comparing results obtained in our study to similar studies, in [42] an average recall of 87% was obtained on the MANIAC data set when using SEC and *assigning cut and chop to the same class*. Though the cited study uses a slightly different procedure (including the class "unknown") and we cannot directly compare, our average classification accuracy in case of also merging the cut and chop classes is 88% in SEC, thus similar to what is achieved in [42]. This confirms that our new similarity measure and our classification procedures, which are different to those in [42], work equally well. Note, the new similarity measure is computationally much more efficient. Furthermore we found that the average ESEC classification accuracy we are obtaining (when merging cut and chop classes) is 97%, thus far higher as compared to that in [42].

6.2. Action prediction

The main focus of this study is action prediction. Thus, we performed a thorough analysis of this using ESECs and we compared this to SECs and also to a baseline method based on hidden Markov models (HMMs). Different from the HMM-based method, both ESEC and SEC frameworks do not require continuous information such as movement trajectories. We compared these three frameworks on two data sets: MANIAC and KIT [47]), resulting in 13 different actions in total (one action type is the same in both data sets).

We showed that on overage the ESEC framework outperforms both SEC and HMM-based methods on both data sets, whereas SEC and HMM methods on average show similar prediction performance. One possible reason for that is that ESEC takes "some middle ground" between SEC and HMM, where SEC, while symbolic, is too compressed and HMM, being sub-symbolic (time continuous), is too prone to noise.

7. Conclusion

In this study, we had presented our augmented action recognition and prediction framework based on ESECs, which is fundamentally based on discrete *events*. Hence, our method does not use continuous (motion trajectory) features or full action reconstruction and performs classification and prediction only by using a symbolic representation relying on the spatial relations between the objects. This way it differs from the great majority of other studies (e.g. [28–30]). As a consequence, our approach has low complexity. Several psychological findings have discussed that event-based encoding might be fundamental for action understanding in the brain (see e.g. [50]) and, thus, our framework might this way indeed be better linked to the way humans "understand" actions. Currently we are pursuing an investigation based on functional magnetic resonance asking whether brain signals will be enhanced in response to such SEC or ESEC events.

Moreover, Given the tight link between NLP and scene description, to further enhance the performance of our approach, we are planning to incorporate NLP (Natural Language Processing) and LfD (Learning from Demonstration) into our framework for future works.

Acknowledgments

The authors are grateful to Ricarda Schubotz and Jennifer Pomp for valuable discussion at several stages of this study. The research leading to these results has received funding from the German Research Foundation (DFG) grant WO388/13-1 and the European Community's H2020 Programme (Future and Emerging Technologies, FET) under grant agreement no. 732266, Plan4Act.

Appendix A. CFG- and PCGF-rules for noise reduction in ESEC

A Context-Free Grammar (CFG) is defined as:

$$G = (N, S, T, P) \tag{8}$$

where N is a finite set of non-terminal symbols, S is the starting symbol ($S \in N$), T is a finite set of terminal symbols ($T \cap N = \emptyset$), P is a finite grammar of the form $A \rightarrow u (A \in N \text{ and } u \in (N \cup T)^{+})$ [51].

As non-terminal symbols we use the symbols defining the relations in the ESECs augmented by the starting symbol st. For definition of the end points of the grammatical transitions, we define a terminal symbol tr. The production rules are obtained according to the manually-created noise-free ESEC matrices of the 35 manipulation actions from Eq. (12). We investigate all possible transitions from each column element to its corresponding next column element and, based on that, generate grammar production rules. Suppose, we want to produce possible transitions from a non-terminal element such as "Moving together" (MT). For this purpose, we search for the element MT in every column of each action $\alpha_i, (1 \leq i \leq 35)$. If it occurs on the R_{th} row and C_{th} column of action α_i ($\alpha_i(R, C) = MT$) then we put the result of the $\alpha_i(R, C + 1)$ into the transition set of MT.

Formally the grammar is described by the following equations:

$$N = \{O, Q, U, T, N, A, X, Ar, Ab, Be, In, Sa, MT, HT, FMT, GC, MA, S\} \tag{9}$$

$$S = st; \tag{10}$$

$$T = \{tr\} \tag{11}$$

$$\begin{aligned}
P = \{ & \\
\bullet St &\rightarrow U|O|Q|N; \\
\bullet T &\rightarrow T|N|A|tr; \\
\bullet N &\rightarrow T|N|A|tr; \\
\bullet A &\rightarrow A|tr; \\
\bullet X &\rightarrow X|tr; \\
\bullet U &\rightarrow U|T|N|X|Ar|Ab|Be|In|Sa|HT|FMT|GC|S|MA|tr; \\
\bullet O &\rightarrow Ab|tr; \\
\bullet Ar &\rightarrow Ar|Ab|O|A|tr; \\
\bullet Ab &\rightarrow Ab|O|In|Ar|X|A|tr; \\
\bullet Be &\rightarrow Be|Ar|tr; \\
\bullet IN &\rightarrow Ab|In|X|tr; \\
\bullet Sa &\rightarrow sa|Ab|tr; \\
\bullet HT &\rightarrow HT|MT|FMT|MA|S|X|Z|tr; \\
\bullet MT &\rightarrow HT|MA|MT; \\
\bullet FMT &\rightarrow MA|HT|FMT|S|X|A; \\
\bullet GC &\rightarrow S|HT|FMT|A|GC; \\
\bullet MA &\rightarrow Q|MA|S|GC|HT; \\
\bullet S &\rightarrow S|Q|MA|GC|FMT|HT|tr; \\
\bullet Q &\rightarrow S|tr. \\
\} &
\end{aligned} \tag{12}$$

The production rules are obtained according to the manually-created noise-free ESEC matrices of the 35 manipulation actions. To get the rules, we analyzed for each non-terminal (which could be a static or dynamic spatial relation) all of these 35 matrices. For each occurrence of a given non-terminal item, the corresponding element of its next column is considered as a possible transition rule of that non-terminal. All of these rules are gathered in Eq. (12). Then, for finding the probability of each transition from a non-terminal to an item, we divided the whole number of transitions from that non-terminal to that item by the total possible transitions from that non-terminal. Suppose, we want to produce possible transitions from a non-terminal element such as "Moving together" (MT). For this purpose, we search for the element MT in every column of each action $\alpha_i, (1 \leq i \leq 35)$. If it occurs on the R_{th} row and C_{th} column of action α_i, ($\alpha_i(R, C) = MT$) then we put the results of the $\alpha_i(R, C + 1)$ onto the transition set of MT. In real data analysis, whenever we faced an illegal transition, we have substituted the transition to the illegal item with the highest possible transition.

Furthermore, we define three types of rules:

1. Rules regarding start point and end point of a manipulation.
2. Rules regarding limitation of transitions for the elements.
3. Rules regarding general consistency of elements in a column.

Regarding the first rule: All of the ESEC matrices for manipulation actions, are started by the symbol: N in the 10 upper rows, U or O in the middle 10 rows and U or Q in the 10 lowest rows.

Regarding the second rule: Only transitions included in the grammar are allowed.

Regarding the third rule: The element should be consistent with the other elements in the same column. If an object is undefined (U), destroyed (X) or absent (A) in the i_{th} row within the set of ten upper rows, $1 \leqslant i \leqslant 10$, the corresponding static spatial relation in rows $(i + 10)_{th}$ and dynamic spatial relations in rows $(i + 20)_{th}$ should be the same. In addition, if two fundamental objects are touching in the i_{th} row, $1 \leqslant i \leqslant 10$, their corresponding dynamic spatial relations in $(i + 20)_{th}$ row should be a member of {HT, MT, FMT}, which are only defined in case of touching and if two object are not touching, their dynamic spatial relation in $(i+20)_{th}$ row should be a member of {MA, GC, FMT, S, Q}. Formally:

$$\begin{aligned}
if(x(i) = X) &\iff x(i + 10) = X \wedge x(i + 20) = X; \\
if(x(i) = A) &\iff x(i + 10) = A \wedge x(i + 20) = A; \\
if(x(i) = U) &\iff x(i + 10) = U \wedge x(i + 20) = U; \\
if(x(i) = T) &\iff x(i + 20) \rightarrow HT|MT|FMT; \\
if(x(i) = N) &\iff x(i + 20) \rightarrow MA|GC|S|Q; \\
&1 \leqslant i \leqslant 10
\end{aligned} \tag{13}$$

In the next paragraphs we define the probabilistic version of the CFG, the so-called PCFG, used for element replacement after detection of a noise event. Similar to the CFG defined above, the PCFG can be defined by a quintuple:

$$G = (N, S, T, P, Pr) \tag{14}$$

where N, S, T, P are compatible with their definitions in the CFG and Pr is the set of probabilities of the production rules. Table 3 shows the production rules (explained in Eq. (12)) with their probabilities based on statistical information obtained from the ESEC matrices of all here investigated manipulations. In PCFG the grammatical rules are given as: $\alpha \rightarrow a_1 * \beta | a_2 * \gamma$. Here a_1 and a_2 are the probabilities of a transition from α to β and γ, respectively.

For example we know according to Eq. (12), the possible transitions from "Around" (Ar) to either the same Ar or "Above" (Ab), or "null" (O), or "Absent" (A). Each of these has a separate probability.[4]

$$Ar \rightarrow a1 * Ar | a2 * Ab | a3 * O | a4 * A \tag{15}$$

According to our available manipulation ESECs, totally there are **213** transitions from Ar, where **169** of them are a transit to Ar, **8** cases go to Ab, **35** elements are converted to O and the remaining

[4] We do not mention tr as a possible transition from Ar because tr is not a legal symbol for our ESEC transitions and is only used as a terminal for ending all transitions.

Table 3
The probability of possible transitions between the spatial relations according to the theoretical statistical analysis of manipulation actions.

	U	T	N	A	X	O	Ar	Ab	Be	In	Sa	Q	HT	FMT	MT	GC	MA	S
U	0.8995	0.0249	0.0085		0.0028		0.0099	0.0205	0.0004	0.0007	0.006		0.0198	0.0038		0.0007	0.0007	0.0066
T		0.8077	0.1813	0.0110														
N		0.0405	0.9558	0.0037														
A				1														
X					1													
O								1										
Ar				0.0047		0.1643	0.7934	0.0376										
Ab				0.0082	0.0049	0.0770	0.0279	0.8738		0.0082								
Be							0.1111		0.8889									
In								0.1111	0.1111	0.7778								
Sa						0.0714		0.2857			0.6429							
Q																		1
HT				0.0074	0.0148								0.6370	0.0556	0.1185		0.1593	0.0074
FMT				0.0417	0.0417								0.3125	0.2917			0.2500	0.0625
MT													0.5517		0.3621		0.0862	
GC				0.0377									0.2642	0.0377		0.0943		0.5660
MA												0.3592	0.0195	0.0032		0.1331	0.2465	0.3521
S												0.1039	0.0195	0.0032		0.1331	0.1461	0.5942

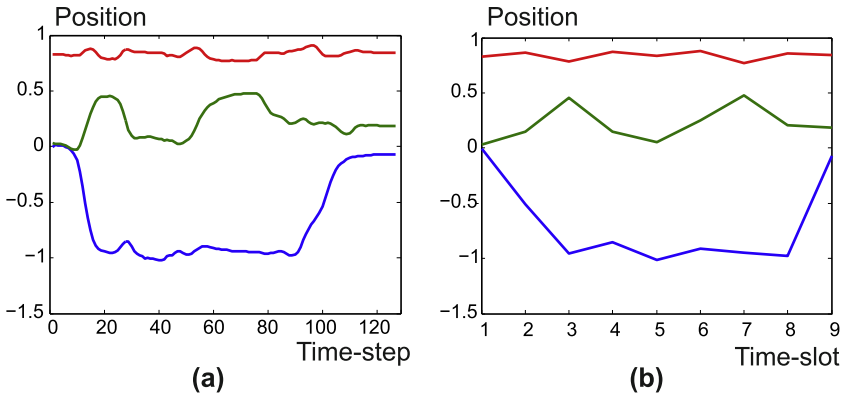


Fig. 11. The effect of Douglas Peucker and DTW pre-processes algorithms on the x, y and z coordinates of a hand trajectory in a cutting demonstration (a): before and (b) after applying the algorithms.

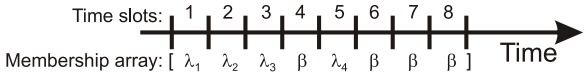


Fig. 12. Membership array of an action during the time.

1 elements changes to *A*. Therefore, **a1**, **a2**, **a3** and **a4** are **0.7934**, **0.0376**, **0.1643** and **0.0047** respectively and above production rule is:

$$Ar \rightarrow 0.7934 * Ar|0.0376 * Ab|0.1643 * O|0.0047 * A \quad (16)$$

These different probabilities can be used to perform column replacement as described in the main text.

Appendix B. HMM-baseline method for hand motion recognition

Here we provide the details of our implementation of the baseline method based on HMMs. We basically follow [40], but expand the method based on the solutions proposed in [46].

Our data relies on hand segments in each video frame. We used the quantization of hand orientation into 18 bins as described by [52]. The obtained codewords were used as input to the HMM recognizer.

Evaluation, decoding and training of the HMM were solved by using *forward–backward algorithm*, *Viterbi algorithm*, and the *Baum–Welch algorithm* respectively, following [40].

Based on the observations in [41,52,53] that – besides *orientation* – also *velocity* and *location* of the hand are useful in hand movement recognition, we have extended the method to include fusion of all these three features. For that we followed the framework from [46].

According to this frameworks, HMMs were fused using majority voting. In addition, we used the Douglas Peucker algorithm and Dynamic Time Warping (DTW) from the same framework for removing noise and hand velocity variations. The effect of the algorithm on our data is shown in Fig. 11.

B.0.1. Extension of recognition to prediction

As the HMM in each time step outputs a recognition result, we have defined that the HMM has predicted the action at that moment from which on recognition remains stable to the end of the action. This approach renders a fair comparison with our ESEC-based prediction.

In more detail: suppose action α shall be truly classified as belonging to manipulation class β. The question is from when the recognition process can yield and keep this result (namely class β as the corresponding class of action α). To achieve this, we analyze time windows. For this, we choose a window size based on the duration of a meaningful motion segment at a medium speed of

the human hand motion. Hence, here we use a window of $\delta = 10$ frames, where the frame-rate in our videos is 30 frames per second. Thus, ten frames is one third of a second and short enough for not missing a meaningful motion. Using this, we divided an action with N frames into $[\frac{N}{\delta}] + 1$ time windows.

Then during the recognition process, the discovered classes of the action α in each time window are found and stored in an array called "membership array", e.g. in Fig. 12 after the first time window the highest membership probability is for class λ_1, after the second time window it will be converted to λ_2 and so on. Then by looking at the obtained membership array, we can conclude from which time window onwards the recognition was stable. In Fig. 12, in the $4_t h$ time window the action is truthfully placed in class β but afterwards it is wrongly recognized as class λ_4. But from the $6_t h$ time window, recognition is correct remains so. Thus, we take the frame θ at the beginning of $6_t h$ time windows or $(5 * \delta + 1)$ frames as the one at which prediction happened. By dividing the frame number of frame θ by the number of total frames and subtracting this from 1 we obtain the predictability of action α.

$$P_F(\alpha) = \left(1 - \frac{\theta}{N}\right) * 100\% \tag{17}$$

which is then used to compare against the performance of the ESEC-based prediction.

References

[1] L.M. Ma, T. Fong, M.J. Micire, Y.K. Kim, K. Feigh, Human-Robot teaming: concepts and components for design, in: Field and Service Robotics, 2018, pp. 649–663.

[2] C. Mandery, Ö. Terlemez, M. Do, N. Vahrenkamp, T. Asfour, The KIT whole-body human motion database, in: Advanced Robotics (ICAR), 2015 International Conference on, 2015, pp. 329–336.

[3] W. Takano, J. Ishikawa, Y. Nakamura, Using a human action database to recognize actions in monocular image sequences: Recovering human whole body configurations, Adv. Robot. 29 (12) (2015) 771–784.

[4] O.D. Lara, M.A. Labrador, A survey on human activity recognition using wearable sensors, IEEE Commun. Surv. Tutor. 15 (3) (2013) 1192–1209.

[5] P. Turaga, R. Chellappa, V.S. Subrahmanian, O. Udrea, Machine recognition of human activities: A survey, IEEE Trans. Circuits Syst. Video Technol. 18 (11) (2008) 1473–1488.

[6] A. Bulling, U. Blanke, B. Schiele, A tutorial on human activity recognition using body-worn inertial sensors, ACM Comput. Surv. 46 (3) (2014) http://dx.doi.org/10.1145/2499621.

[7] E.E. Aksoy, Y. Zhou, M. Wächter, T. Asfour, Enriched manipulation action semantics for robot execution of time constrained tasks, in: IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids), 2016, pp. 109–116.

[8] C. Fermüller, F. Wang, Y. Yang, K. Zampogiannis, Y. Zhang, F. Barranco, M. Pfeiffer, Prediction of manipulation actions, Int. J. Comput. Vis. 126 (2–4) (2018) 358–374.

[9] J. Dinerstein, D. Ventura, P.K. Egbert, Fast and robust incremental action prediction for interactive agents, Comput. Intell. 21 (1) (2005) 90–110.

[10] Y. Kong, Z. Tao, Y. Fu, Deep sequential context networks for action prediction, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1473–1481.

[11] F. Ziaeetabar, E.E. Aksoy, F. Wörgötter, M. Tamosiunaite, Semantic analysis of manipulation actions using spatial relations, in: IEEE Int. Conf. on Robotics and Automation, ICRA, 2017, pp. 4612–4619.

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

[13] T.M. Crockett, M.W. Powell, K.S. Shams, Spatial planning for robotics operations, in: IEEE Aerospace Conference, 2009, pp. 1–7.

[14] J.A. Park, Y.S. Kim, J.Y. Cho, Visual reasoning as a critical attribute in design creativity, in: International Design Research Symposium, 2006.

[15] M. Cantero, F. Naya, J.L. Saorín Pérez, et al., Learning Support Tools for Developing Spatial Abilities in Engineering Design, 2006.

[16] S. Eagleson, F. Escobar, I. Williamson, Hierarchical spatial reasoning theory and GIS technology applied to the automated delineation of administrative boundaries, Comput. Environ. Urban Syst. 26 (2–3) (2002) 185–200.

[17] Y. Wei, E. Brunskill, T. Kollar, N. Roy, Where to go: Interpreting natural directions using global inference, in: IEEE Int. Conf. on Robotics and Automation, ICRA, 2009, pp. 3761–3767.

[18] G. Gemignani, R. Capobianco, D. Nardi, Approaching qualitative spatial reasoning about distances and directions in robotics, in: Congress of the Italian Association for Artificial Intelligence, 2015, pp. 452–464.

[19] J.K. Aggarwal, Q. Cai, Human motion analysis: a review, Comput. Vis. Image Underst. 73 (3) (1999) 428–440.

[20] J.W. Davis, A.F. Bobick, The representation and recognition of human movement using temporal templates, in: IEEE Conf. on Computer Vision and Pattern Recognition, CVPR, 1997, pp. 928–934.

[21] T.B. Moeslund, E. Granum, A survey of computer vision-based human motion capture, Comput. Vis. Image Underst. 81 (3) (2001) 231–268.

[22] C. Wallraven, B. Caputo, A. Graf, Recognition with local features: the kernel recipe, in: IEEE Int. Conf. on Computer Vision, ICCV, 2003, http://dx.doi.org/10.1109/ICCV.2003.1238351.

[23] T. Lan, L. Sigal, G. Mori, Social roles in hierarchical models for human activity recognition, in: IEEE Conf. on Computer Vision and Pattern Recognition, CVPR, 2012, pp. 1354–1361.

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

[25] A. Patron-Perez, M. Marszalek, I. Reid, A. Zisserman, Structured learning of human interactions in tv shows, IEEE Trans. Pattern Anal. Mach. Intell. 34 (12) (2012) 2441–2453.

[26] B. Yao, X. Jiang, A. Khosla, A.L. Lin, L. Guibas, L. Fei-Fei, Human action recognition by learning bases of action attributes and parts, in: IEEE Int. Conf. on Computer Vision, ICCV, 2011, pp. 1331–1338.

[27] M.S. Ryoo, Human activity prediction: early recognition of ongoing activities from streaming videos, in: IEEE Int. Conf. on Computer Vision, ICCV, 2011, pp. 1036–1043.

[28] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J.M. Siskind, S. Wang, Recognize human activities from partially observed videos, in: IEEE Conf. on Computer Vision and Pattern Recognition, CVPR, IEEE, 2013, pp. 2658–2665.

[29] T. Lan, T.C. Chen, S. Savarese, A hierarchical representation for future action prediction, in: Europ. Conf. on Computer Vision, ECCV, 2014, pp. 689–704.

[30] H.S. Koppula, A. Saxena, Anticipating human activities using object affordances for reactive robotic response, IEEE Trans. Pattern Anal. Mach. Intell. 38 (1) (2016) 14–29.

[31] Z. Wang, M.P. Deisenroth, H.B. Amor, D. Vogt, B. Schölkopf, J. Peters, Probabilistic modeling of human movements for intention inference, Proc. Robot. Sci. Syst. VIII (2012).

[32] M. Sridhar, A.G. Cohn, D.C. Hogg, Learning functional object categories from a relational spatio-temporal representation, in: Europ. Conf. on Artificial Intelligence, ECAI, 2008, pp. 606–610.

[33] K. Ramirez-Amaro, E.S. Kim, J. Kim, B.T. Zhang, M. Beetz, G. Cheng, Enhancing human action recognition through spatio-temporal feature learning and semantic rules, in: IEEE-RAS Int. Conf. on Humanoid Robots, Humanoids, 2013, pp. 456–461.

[34] K. Ramirez-Amaro, M. Beetz, G. Cheng, Extracting semantic rules from human observations, in: ICRA Workshop: Semantics, Identification and Control of Robot-Human-Environment Interaction, 2013.

[35] E.E. Aksoy, A. Orhan, F. Wörgötter, Semantic decomposition and recognition of long and complex manipulation action sequences, Int. J. Comput. Vis. 122 (1) (2017) 84–115.

[36] K. Nagahama, K. Yamazaki, K. Okada, M. Inaba, Manipulation of multiple objects in close proximity based on visual hierarchical relationships, in: IEEE Int. Conf. on Robotics and Automation, ICRA, 2013, pp. 1303–1310.

[37] Y. Yang, C. Fermüller, Y. Aloimonos, A cognitive system for human manipulation action understanding, in: Ann. Conf. on Advances in Cognitive Systems, ACS, vol. 2, 2013.

[38] D.R. Faria, R. Martins, J. Lobo, J. Dias, Extracting data from human manipulation of objects towards improving autonomous robotic grasping, Robot. Auton. Syst. 60 (3) (2012) 396–410.

[39] K. Zampogiannis, Y. Yang, C. Fermüller, Y. Aloimonos, Learning the spatial semantics of manipulation actions through preposition grounding, in: IEEE Int. Conf. on Robotics and Automation, ICRA, 2015, pp. 1389–1396.

[40] M. Elmezain, A. Al-Hamadi, B. Michaelis, Hand gesture recognition based on combined features extraction, World Acad. Sci. Eng. Technol. 60 (2009) http://dx.doi.org/10.1999/1307-6892/1761.

[41] M. Elmezain, A. Al-Hamadi, B. Michaelis, Hand trajectory-based gesture spotting and recognition using hmm, in: IEEE Int. Conf. on Image Processing, ICIP, 2009, pp. 3577–3580.

[42] E.E. Aksoy, M. Tamosiunaite, F. Wörgötter, Model-free incremental learning of the semantics of manipulation actions, Robot. Auton. Syst. 71 (2015) 118–133.

[43] 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 Trans. Auton. Mental Dev. 5 (2) (2013) 117–134.

[44] J. Papon, A. Abramov, M. Schoeler, F. Wörgötter, Voxel cloud connectivity segmentation-supervoxels for point clouds, in: IEEE Conf. on Computer Vision and Pattern Recognition, CVPR, 2013, pp. 2027–2034.

[45] A. Abramov, K. Pauwels, J. Papon, F. Wörgötter, B. Dellen, Depth-supported real-time video segmentation with the kinect, in: IEEE Workshop on Applications of Computer Vision, WACV, 2012, pp. 457–464.
[46] J. Singha, R.H. Laskar, Hand gesture recognition using two-level speed normalization, feature selection and classifier fusion, Multimedia Syst 23 (4) (2017) 499–514.
[47] C. Mandery, Ö. Terlemez, M. Do, N. Vahrenkamp, T. Asfour, Unifying representations and large-scale whole-body motion databases for studying human motion, IEEE Trans. Robot. 32 (4) (2016) 796–809.
[48] M. Aiello, B. Ottens, The mathematical morpho-logical view on reasoning about space, in: IJCAI, 2007, pp. 205–211.
[49] F. Ziaeetabar, T. Kulvicius, M. Tamosiunaite, F. Wörgötter, Prediction of manipulation action classes using semantic spatial reasoning, in: IEEE Int. Conf. on Intelligent Robots and Systems, IROS, 2018, (in press).
[50] B. Hommel, The theory of event coding (tec) as embodied-cognition framework, Front. Psychol. 6 (2015) 1318.
[51] A. Rosani, N. Conci, F.G. De Natale, Human behavior recognition using a context-free grammar, J. Electron. Imaging 23 (3) (2014) http://dx.doi.org/10.1117/1.JEI.23.3.033016.
[52] M. Elmezain, A. Al-Hamadi, J. Appenrodt, B. Michaelis, A hidden markov model-based continuous gesture recognition system for hand motion trajectory, in: Int. Conf. on Pattern Recognition, ICPR, 2008, pp. 1–4.
[53] C.Y. Kao, C.S. Fahn, A human-machine interaction technique: hand gesture recognition based on hidden Markov models with trajectory of hand motion, Procedia Eng. 15 (2011) 3739–3743.

Fatemeh Ziaeetabar is a Ph.D. student in computer science at Göttingen University. Her current research interests are machine vision, semantic understanding, robotics, artificial intelligence and machine learning.

Tomas Kulvicius received his Ph.D. degree in Computer Science (2010) from the University of Göttingen, Germany. In his Ph.D. thesis he investigated development of receptive fields in closed loop learning systems. From 2010 to 2015, he was a researcher at the University of Göttingen where he worked on trajectory generation and motion control for robotic manipulators. From 2015 to 2017, he was appointed as an Assistant Professor at the Centre for Bio Robotics, University of Southern Denmark. Currently he is a Research Assistant at the University of Göttingen, Germany. His research interests include modeling of closed-loop behavioral systems, robotics, artificial intelligence, machine learning algorithms, movement generation and trajectory planning.

Minija Tamosiunaite has received a Ph.D. in Informatics in Vytautas Magnus University, Lithuania, in 1997. Currently she works as a senior researcher at the Bernstein Center for Computational Neuroscience, Inst. Physics 3, University of Göttingen. Her research interests include machine learning, biological signal analysis, and application of learning methods in robotics.

Florentin Wörgötter has studied biology and mathematics at the University of Düsseldorf, Germany. He received the Ph.D. degree for work on the visual cortex from the University of Essen, Germany, in 1988. From 1988 to 1990, he was engaged in computational studies with the California Institute of Technology, Pasadena, CA, USA. Between 1990 and 2000, he was a Researcher at the University of Bochum, Germany, where he was investigating the experimental and computational neuroscience of the visual system. From 2000 to 2005, he was a Professor of computational neuroscience with the Psychology Department, University of Stirling, U.K., where his interests strongly turned towards Learning in Neurons. Since July 2005, he has been the Head of the Computational Neuroscience Department at the Bernstein Center for Computational Neuroscience, Inst. Physics 3, University of Göttingen, Germany. His current research interests include information processing in closed-loop perception action systems, sensory processing, motor control, and learning/plasticity, which are tested in different robotic implementations.

Chapter 5

Manipulation Action Prediction By Virtual Reality: A Comparison Between Human and ESEC Predictability Power

5.0.1 Motivation

The main topic of this thesis is about prediction of manipulation actions. For humans, this is generally an easy task. Most activities can already be identified by recognition of the participating objects or movement of the hand before the activity started. This raises the question of how important spatial relations between objects are to the human understanding of actions and how humans compete with computer vision algorithms when all manipulation action objects are represented as cubes of random size and color. Hence, when all helpful object information is removed. To address this issue we performed a large set of psychophysics experiment in a virtual reality setup.

5.0.2 Introduction

Virtual reality (VR) replaces boring, flat monitors with 3D worlds that immerse users in unique visual experience. The concept of VR has been around since about 1970. It involves cutting edge devices that totally engross the user's vision system and increase their cognitive awareness of 3D scenarios. VR navigation means that users experience themselves manipulating objects, reacting to events and moving around and exploring landscapes. This special immersion is achieved by using real-time, stereoscopic sound and graphics that present virtual worlds in first-person views. In fact, VR technology provides intuitive ways for users to explore new environments and master new skills.

VR can be found in fields as diverse as entertainment, marketing, education, medicine, construction, road safety training, and many others. They provide numerous possibilities for users to explore virtual realities for various purposes [50] [51].
Moreover, VR tools give science a new dimension and allow scientific researchers to view and share data as never before. As an example, virtual reality is being increasingly used in the field of scientific visualization. This field is based on using computer graphics to express complex ideas and scientific concepts like molecular models or statistical results. Scientific visualization is used as a means of communicating abstract concepts to an audience which also aids understanding. The audience can interact with these images, for example, viewing a molecular structure at different angles or as a means of problem solving [52]. Virtual reality enables scientists to demonstrate a method or convey complex ideas in a 3D, interactive visual format.

Virtual Reality (VR) is not an entirely new concept; it has existed in various forms since the late 1960s. It has been known by names such as synthetic environment, cyberspace, artificial reality, simulator technology and so on and so forth before VR was eventually adopted. The latest manifestation of VR is desktop VR [53]. VR promised a new concept of technology interaction, different from traditional mouse and keyboard input with stationary monitors. After several

attempts to develop consumer products failed, the technology began to gain traction and huge public interest again, when Oculus VR announced the Oculus Rift headset in 2012. Since then, many hardware manufactures invested in their own VR headsets and controller development.

Our goal in this chapter of the thesis is to use virtual reality to test our framework. For this, we defined 10 manipulation actions and made 30 sample scenarios for each of them (each scenario with different objects and different scene arrangements). After that, we arranged these 300 generated scenarios as random sequences. Then we conducted a test on 50 people. The test routine is that for each person, we show 300 scenarios with an interval of 1 second in random order and we ask them to inform the system by clicking on a button as soon as the type of manipulation has been recognized.

The moment and accuracy of the predictions are stored for each trial and the participants' predictability power is calculated accordingly.

We also apply our ESEC framework on the same 300 sample scenarios and finally we made a comparison between the predictability power in humans and in the ESEC framework. We had already compared the algorithmic ESEC framework with SECs and HMMs and this new set of experiments now does the same with human prediction performance.

5.0.3 Outline

We will first explain the virtual reality system that we used. Afterwards, we demonstrate the methods employed in our VR experiments and then we describe the results of these experiments and compare them with the results of the ESEC theoretical analysis.

5.0.4 Virtual Reality System

"We are finally going to be free of 2D monitor. It has been a window into virtual reality that we have all looked into for 30 or 40 years."
Brendan Iribe

Key Components in a Virtual Reality System

The general components necessary for building and experiencing VR are sub-divided as listed below.

- **PC (Personal Computer)/Console/Smart phone**
  Virtual reality content, which is what users view inside of a virtual reality headset, is very data-rich. In order to power these interactive three-dimensional environments, significant

computing power is required. This is where PC (Personal Computer), consoles, and smart phones come in. They act as the engine to power the content being produced (Fig.5.1 (a)).

- **Head-Mounted Display**
  A head-mounted display (also called HMD, Headset, or Goggles) is a type of device that contains a display mounted in front of a user's eyes. This display usually covers the user's full field of view and displays the virtual reality content (Fig.5.1 (b)).
- **Input Devices**
  Input devices or controllers are one of the two categories of components that provide users with a sense of immersion (i.e. convincing the human brain to accept an artificial environment as real). They provide users with a more natural way to navigate and interact within a virtual reality environment (Fig.5.1 (c)).

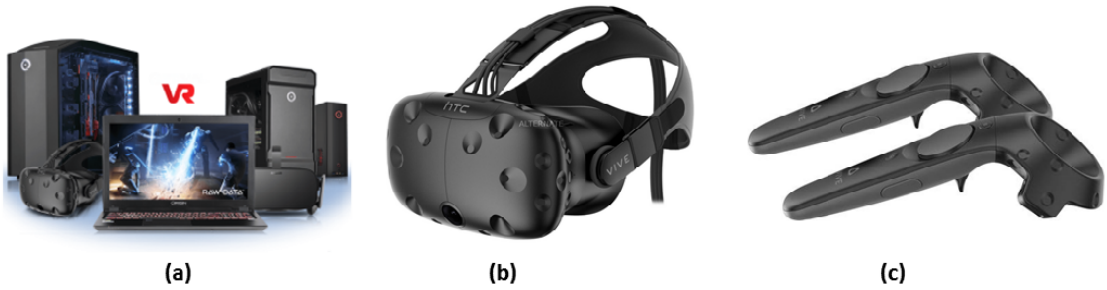


Figure 5.1: VR main components: (a) Computing power, (b) Head-Mounted Display, (c) Motion controllers

Choice of Development Environment

The Vive VR headset and controller are used in this project. It was released by HTC in April 2016 and features a resolution of 1080x1200 per eye. Its main advantage over competing headsets is its "roomscale" system, which allows for precise 3D motion tracking in between two infrared base stations. This provides the opportunity to record and review actions for the experiment on a larger scale of up to 5 meters diagonally. Furthermore, it is widely supported by all VR game engines.

The Unreal Engine 4 (UE4) is a high performance game engine developed by Epic Games and is chosen as the game engine of this project. It has built-in support for virtual reality environments and the Vives motion controllers. After registering and linking an Epic Games account with a GitHub account, source code access is granted, which greatly helps in development.
The virtual reality system used in this project was designed and implemented as a bachelor's thesis by "Stefan Pfeiffer" [54].

5.0.5 Virtual Reality Experiment

Action Types

The first step was defining an action set. These 10 actions were chosen for the experiment:

- Chop
- Cut
- Hide
- Uncover
- Put on top
- Take down
- Lay
- Push
- Shake
- Stir

All objects, including hand and tools, in all actions are represented by colored cubes of variable size, color, and location. This is done to allow for a fair comparison with the ESEC method, because our ESEC framework does not use any object recognition method. Hence, we decided to design our experiment in such a way that the type of manipulated objects does not provide guidance to the type of actions. The hand, which is the most important object in a manipulation, is always shown as a red cube.

Chop: The hand-object (short: hand) touches an object (tool), picks up the object from the ground, puts it on another object (target) and starts chopping. When the target object was divided into two parts, the tool object untouches the pieces of the target object. After that, the hand puts the tool object on the ground, untouches it, and leaves the scene.

Cut: The hand touches an object (tool), picks up the object from the ground, puts it on another object (target) and starts cutting. When the target object was divided into two parts, the tool object untouches the pieces of the target object. After that, the hand puts the tool object on the ground, untouches it, and leaves the scene.

Hide: The hand touches an object (tool), picks up the object from the ground, puts it on another object (target) and starts coming down on the target object until it covers that object thoroughly. Then the hand untouches the tool object and leaves the scene.

Uncover: The hand touches an object (tool), picks up the object from the ground. The second object (target) emerges as the tool object is raised from the ground, because the tool object had hidden the target object. After that, the hand puts the tool object on the ground, untouches it, and leaves the scene.

Put on top: The hand touches an object, picks up the object from the ground and puts it on another object. After that, the hand untouches the first object and leaves the scene.

Take down: The hand touches an object that is on another object, picks up the first object from the second object and puts it on the ground. After that, the hand untouches the first object and leaves the scene.

Lay: The hand touches an object on the ground and changes its direction (lays it down) while it remains touching the ground. After that, the hand untouches the object and leaves the scene.

Push: The hand touches an object on the ground and starts pushing it on the ground. After that, the hand untouches the object and leaves the scene.

Shake: The hand touches an object, picks up the object from the ground and starts shaking it. Then, the hand puts it back on the ground, untouches is, and leaves the scene.

Stir: The hand touches an object (tool), picks up the object from the ground, puts it on another object (target) and starts stirring. After that, the hand puts the tool object on the ground, untouches it, and leaves the scene.

For each action 30 samples were recorded. As an important point, the action scenes should never be distinguishable at the start. Imagine a scene based on the action set with only two visible cubes at its beginning, with one being the hand. Most actions could be ruled out immediately, as they require the second, to be picked up object, to interact with a third one. This leaves only *shake*, *push* and *uncover* as options. Therefore, it was necessary to design our sample scenarios in a way that no one can predict the type of actions from the scene arrangements. Hence, always many blocks are shown at the start of an experiment.

Experiment Process

A view of a Vive motion controller and its buttons are shown in the fig.5.2. The experiment is automatically started and completely controlled by the subject through pressing 5/Down (fig.5.2) to advance to the next state. This subsection therefore focuses on the experimental procedures and gives visual samples of the different stages.

As human performance system test, **50** people were recruited for the experiment.Among of them, **35** persons were **male** and **15** persons were **female**. The **youngest** person was **20** and the **oldest** one was **68** years old. The **average age** of participants was **31.62** and their **median age** was **29**.
All people were given an introduction and explanation of the goal and instructed to press the

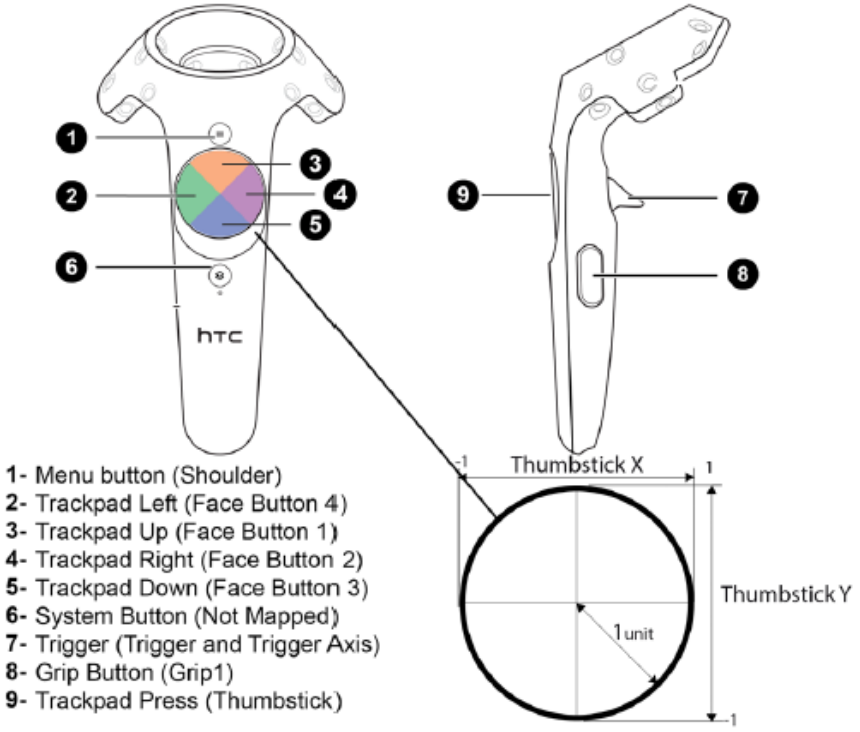


Figure 5.2: Vive Motion Controller Buttons

button immediately after realizing which action is being performed in every step. The resulting (50*300=15000) answers are separated by action and subject number. Participant performance is then evaluated, by dividing the time at which the answer was given by the total duration of the scene. This measure is called *"Human predictability Power"* and describes how much (in percent) of the scene had been seen by the participant before a choice was made.

Before each experiment begins, in a training stage an example of each action (10 samples in total) is displayed to the participant to show them how each action is performed with cubes. There is always a list of actions in the back of these training scenes and, during the display of each action, the cell containing its name is shown in green. Fig.5.3 demonstrates an example of a "put on top" action in the training stage.

After the end of the training stage, the test stage begins. The red hand-cube enters the scene, picks up a cube, and performs an action (fig.5.4). When the action is recognized and the participant presses the 5/Down button and the moment of this butto-press is recorded as the reaction time. At that moment also all cubes are removed from the scene so that no post-decision cogitation about the scene is possible. The controller gets a red pointer added to its front. Hovering over the action of choice and pressing 5/Down again records the actual choice and advances the experiment to

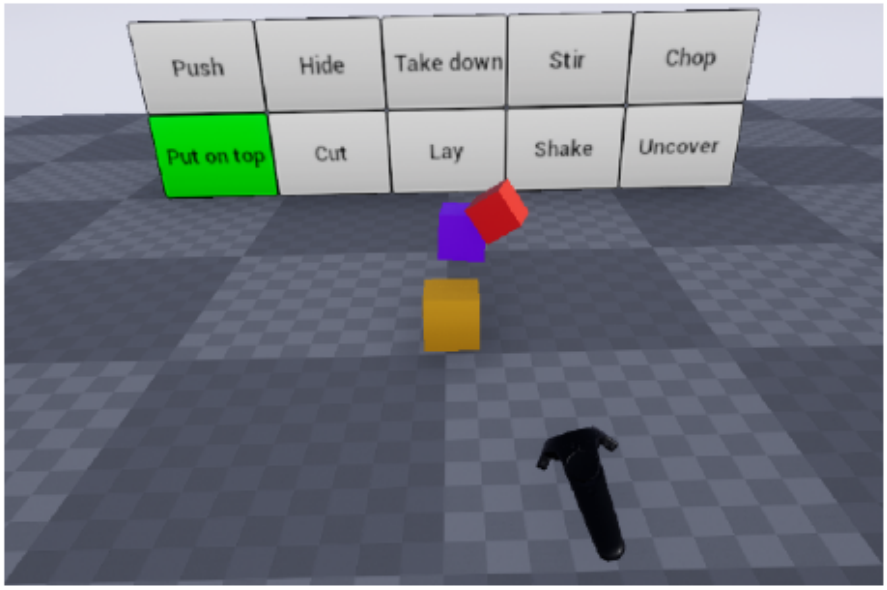


Figure 5.3: Experiment Training Stage: Put on top action

the next trial (fig.5.5).

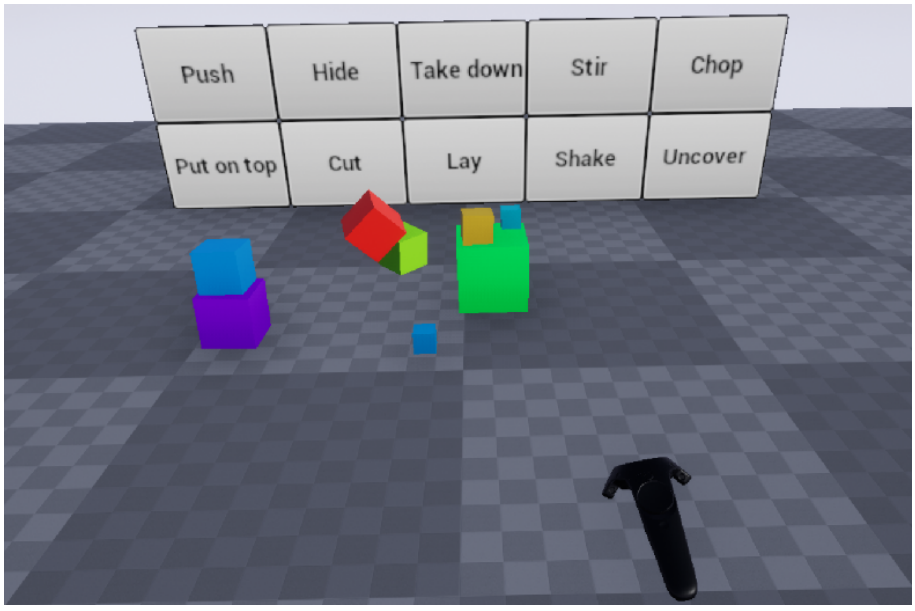


Figure 5.4: Experiment Testing Stage: Action scene playing

At the beginning of the experiment, a result file will be created. Every answer appends a line to the file, indicating whether the answer was correct or not, the participant reaction time in seconds, the

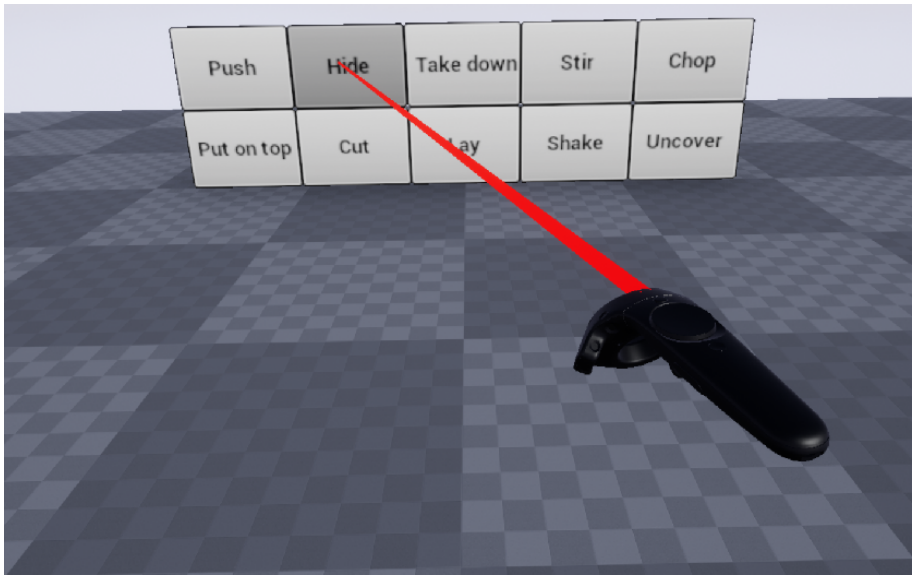


Figure 5.5: Experiment Training Stage: Choose the action

name of the action type that the participant has predicted plus the correct name of the action type. The last item in this line is the name of the recording action file shown to the participant.

This file includes 300 lines (the number of trials) for each person experiment. One of these lines sample is shown in fig.5.6.

ANSWER: correct: yes, **time:** 13.827798, **guess:** Chop, **action:** Chop, **file:** exp/chop/2018-07-06_15-46-16.recording.json

Figure 5.6: Experiment Result File Format

5.0.6 Results

As we discussed in the last section, we selected 10 manipulations and recorded 30 different samples of each, thus generating 300 sample scenarios. We did both theoretical(ESEC based) and human experimental analysis on this data and did a proper comparison between them. Now we want to explain both in more detail.

ESEC Framework Results

We performed Monte Carlo cross validation 20 times, where each time we randomly selected 20 different actions from each class (in total 10*20=200 actions) for training and used them as action

Table 5.1: Average and Median of predictability power for all action types according to the ESEC framework

Manipulation type	Average	Median
Chop	36.55%	35.78%
Cut	52.11%	51.25%
Hide	36.35%	36.88%
Uncover	55.51%	55.86%
Put on top	19.61%	21.16%
Take down	59.31%	58.39%
Lay	50.70%	50.57%
Push	61.71%	60.81%
Shake	46.09%	45.71%
Stir	55.82%	55.61%
Total	**47.38%**	**50%**

models for comparison and 10 actions from each class (in total 10*10=100 actions) for testing. Here, the *"Train-Test Ratio"* was considered as 66.66%, which is a usual ratio in such studies. The prediction process is exactly same with the method, which was described before for *"frame based prediction"*. As its result, predictability power of each sample scenario based on ESEC framework is computed. Predictability power=100% means immediate prediction (impossible) and Predictability power=0% means prediction happened exactly when the displayed action has ended on the screen.

The average and median values of each action typeś predictability power are shown in table.5.1. As a result of this table, we conclude that the ESEC framework mode can on average predict the type of action after **52.62%** of its progress. According to the total median value, it can make a correct prediction exactly in the middle **50%** of the actions. The detailed enriched semantic event chain matrices — with the event columns indicating the prediction place in the theoretical analysis as well as humans — are explained in the section 5.0.8.

Human results

- **Removing Low-Performer Data** The first point, which should be considered before the analysis of human results, is removing possible data points of human, which substantial below-average accuracy in action recognition. Fig.5.7 is a plot showing the relationship between predictability power (speed of prediction) and accuracy (number of wrong recognitions) in the 50 examined people. This figure also includes the data linear fit ($y = 0.0926x + 29.8507, R\_value = 0.1380$).

  As can be seen, there is only one person (rightmost data point) who, in comparison with the rest, produces a significant number of mistakes. Therefore, we will remove this person in all subsequent analyzes and work with the other 49 participant after this. According to a linear fit and its positive gradient, the people with higher predictability power usually (but not much) make more mistakes. Likely, this is because faster people do not wait until they are fully sure of

their predictions and only decide on the initial evidences.

In the following, we show data and discuss potential "learning effects" and "variability of predictability power" of the participants in all 10 actions of the VR experiment.

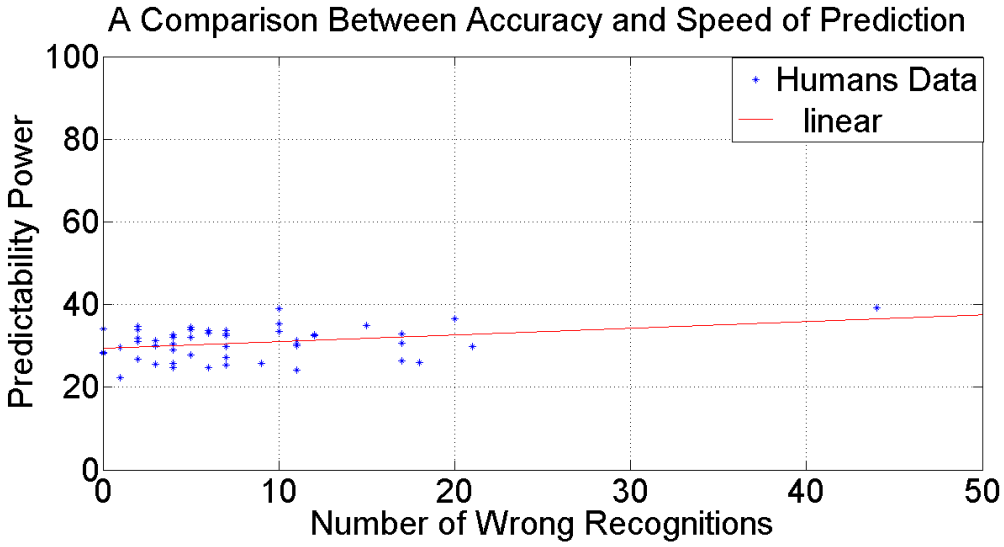


Figure 5.7: Comparison between speed and accuracy in the VR action prediction experiment.

- **Learning Effects**
  One important question that arises is whether people show learning or performance improvement during the trials?
  There are 30 trial scenarios for each manipulation in the VR test. Thus, we want to know if the participants' predictability power for each action type increases by seeing more examples of that action during the experiment. In other words, does a person's prediction power in the first samples show a significant difference as compared to the last examples? To answer this question, for each manipulation action type, we show a bar plot whose horizontal axis is a number from 1 to 30 that represents the trial number and the vertical axis shows the average predictability power of individuals for that trial. Also, we plot the value of standard deviation (STD) on each bar. Fig.5.8 shows the learning effect among of the participants on all 10 discussed manipulations. Finally, Fig.5.9 illustrates the effect of learning on the predictability power of the participants across the grant-average of all 10 actions.
  As can be deduced from the Fig.5.9, in the average mode, a very small learning effect is only observed in the first 5 trials, which, however, is not true for some actions like hide and take (Fig.5.8). All in all, learning is *not significant* along trials.

- **Variability of Predictability Power in the Same Action**
  Another interesting question to be asked here is the comparison of the predictability power of the participants about a specific action. In fact, we want to know: are there different strategies

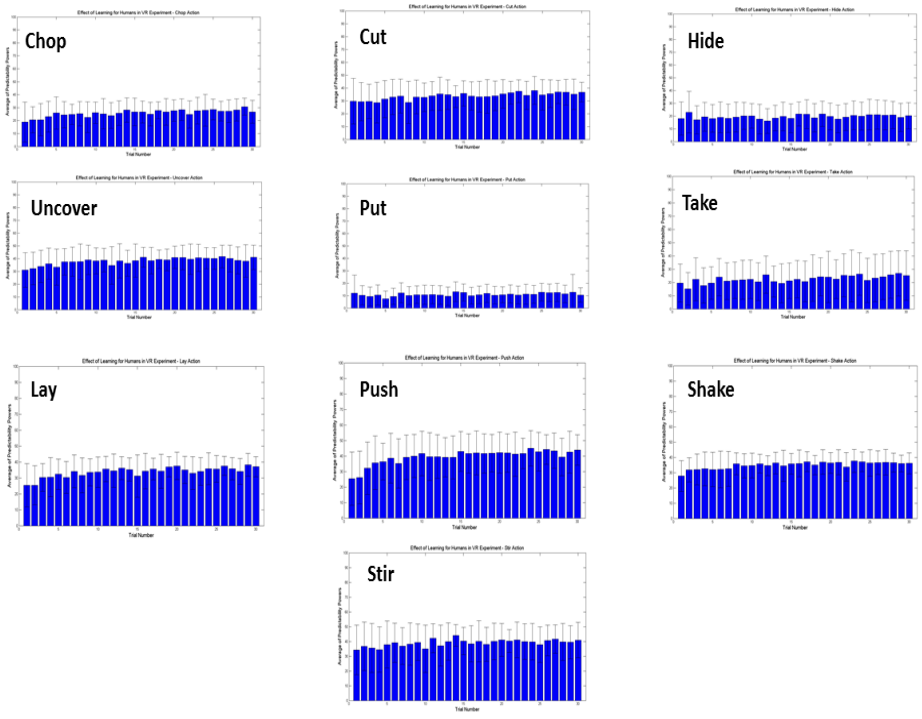


Figure 5.8: Increasing the number of observations and its effect on improving the average predictability power of the participants for each of the 10 manipulations

that may lead to some people performing faster and others slower at recognizing the actions. Consequently, for each of the 10 manipulation actions, we extracted histograms (Fig.5.10), whose horizontal axis shows the median values of the predictability power of individuals (the bin-width of the histogram bars was considered as 3) and their vertical axis is the number of individuals that have shown this particular predictability. More details of one of these histograms, are shown in Fig.5.11, which is the enlarged histogram of the *"shake"* action in Fig.5.10. According to this figure the median of predictability power values of all individuals to recognize the *shake* action is different, while the range of these numbers is between 21 and 42, the largest number is in range 36 to 39. It means, although some people are slower and a few are faster, but most of them (31 out of 49 people) predict this action with the predictability power from 36 to 39.

These figures are important because they show the predictability power of the participants in the same manipulation is different, and the distribution of this ability is various from action to action, as well. For example, in some actions like *shake* and *uncover*, more people perform better than others, while in *put on top* action, most people have poor performance and lower predictability power. Also, in actions like *cut* and *chop*, there is no distinct majority with better or worse results and they contain a distribution of individuals in a wide range of predictability

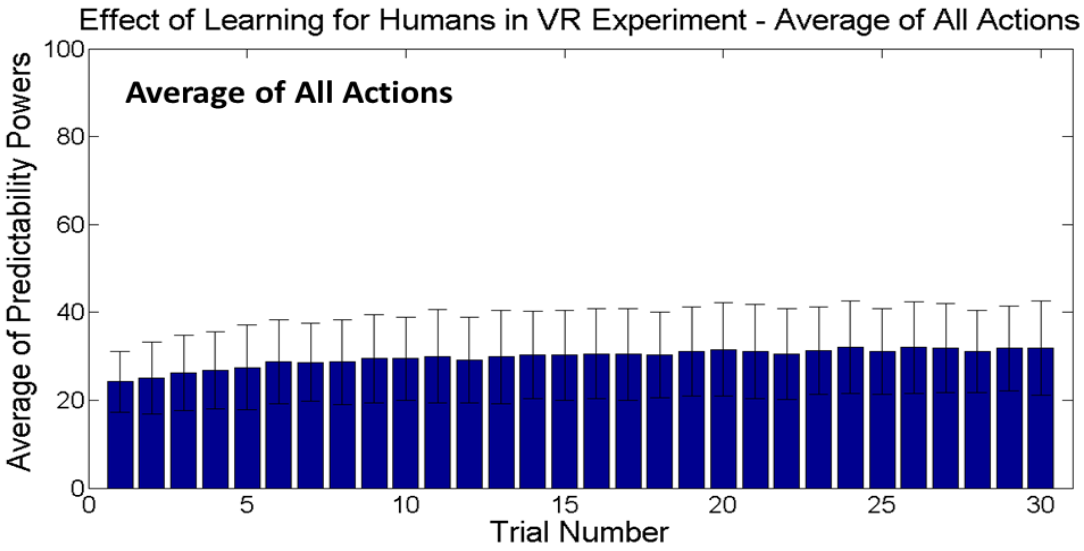


Figure 5.9: Learning effect on improving the average predictability power of the participants for the average of all manipulations

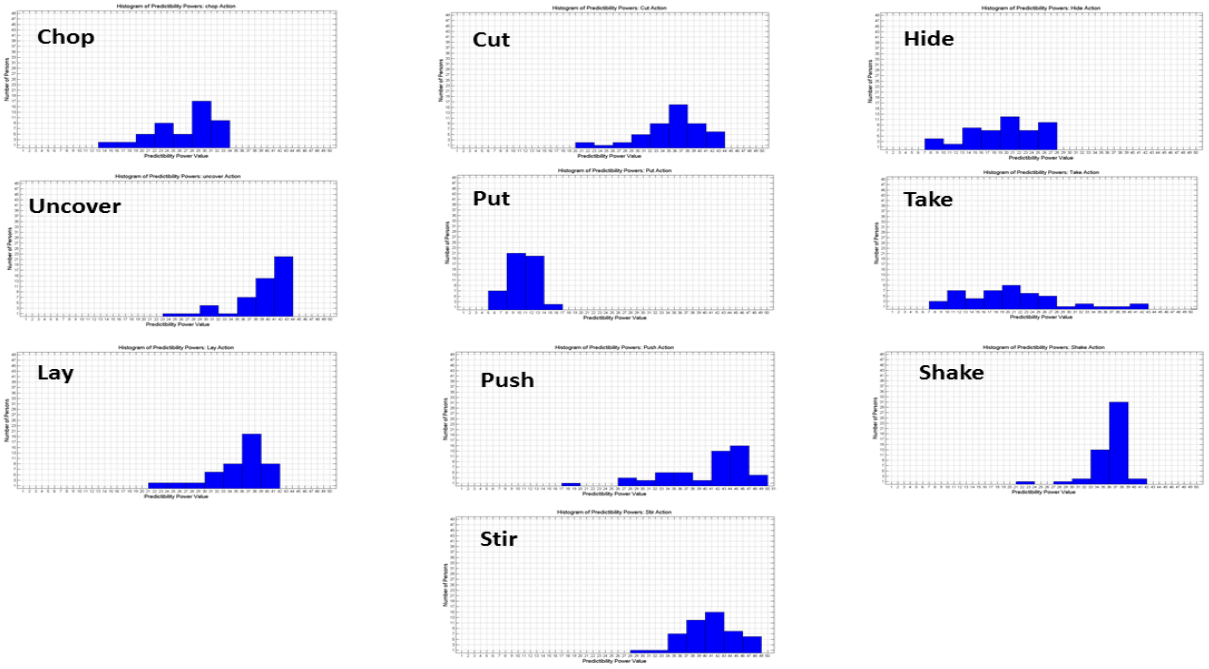


Figure 5.10: Histograms of the median predictability power of the participants for all 10 manipulations

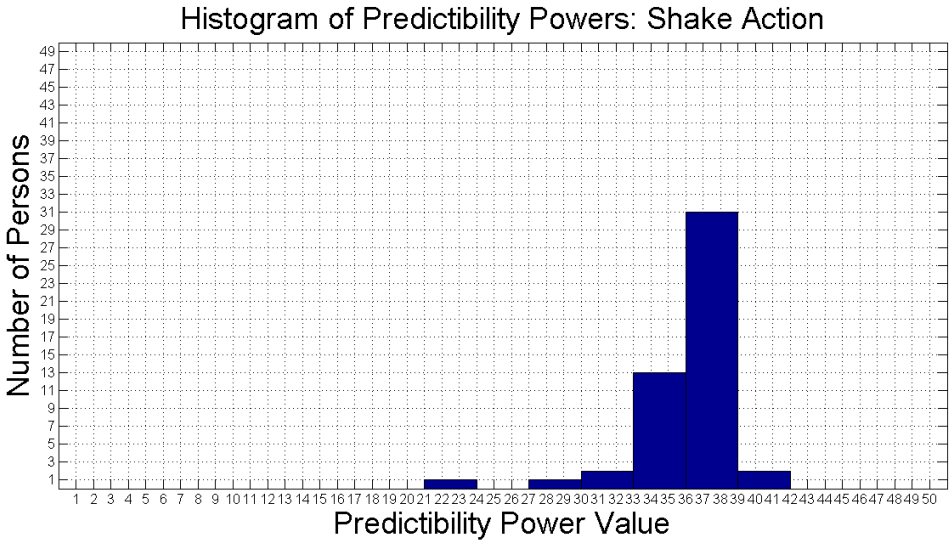


Figure 5.11: Histogram of the median predictability power of the participants for the *shake* action

powers.

In the next step, we want to compare the predictability power results of the participants with our ESEC algorithm and see how similar is the column of the ESECs matrix in which the prediction of the algorithm takes place and the column in which the majority of people make prediction.

5.0.7 Comparison Between ESEC Framework and Human Performance

In this section we want to compare results from the ESEC framework to those from humans and answer the question: **"Are humans performing faster or slower than the ESEC framework?"**

We know that, when our ESEC algorithm collects enough evidence to predict the type of action, the system is informed immediately, but a human must press a key on the VR controller and inform the system about the predicted action. The length of time taken for a person to respond to a given stimulus or event is called **"motor reaction time"** and to have a fair comparison, we have to deduct the motor reaction time from the total reaction time used for a prediction. We have considered this parameter as 300 milliseconds based on the [55] and [56] studies. According to these studies reaction time depends on a number of external (stimulation intensity, sensory modality, sensory quality of signal, pulse-to-pulse interval, etc.) and internal (age, gender, professional skill, functional state, etc.) factors which in average mode is considered to be 300 milliseconds for simple task like pressing a button in our VR experiment.

Fig.5.12 and fig.5.13 compare the predictability power between ESEC and our 49 praticipants without and with consideration of the motor reaction time, respectively. Both plots are divided into

10 parts, each part is assigned to a specific action type and the dots represent the median of each person's predictability power for that action. The green horizontal lines and the red vertical bars show the median of all participants' predictability power and the median of the ESEC algorithm for the related action, respectively.

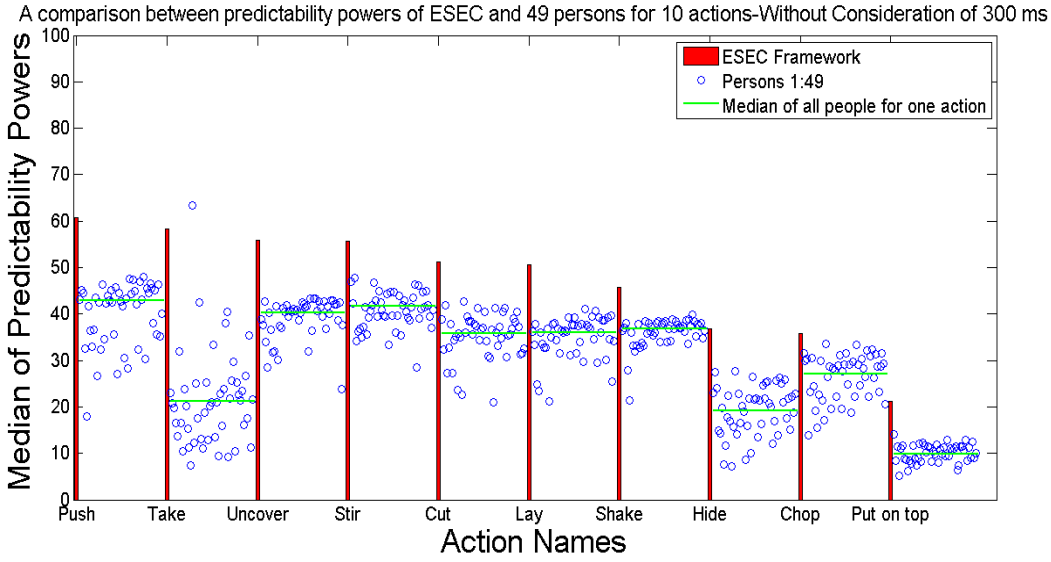


Figure 5.12: Comparison of the ESEC and the participants' median predictability power without consideration of the reaction time

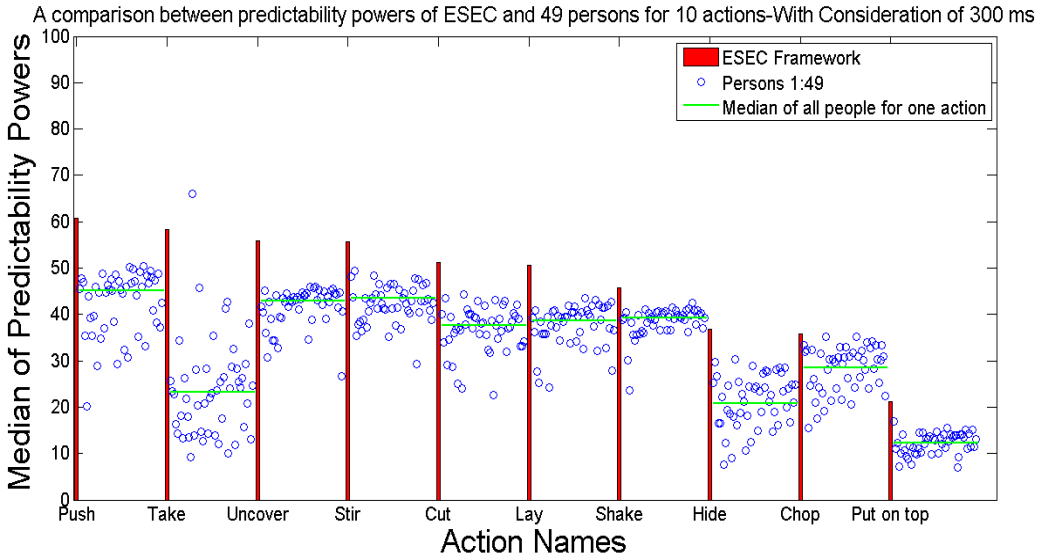


Figure 5.13: Comparison of the ESEC and the participants' median predictability power with consideration of the reaction time

As can be seen from these diagrams, reaction time correction shifts the data points and the medians a bit upward, but the shift value is not much as the trials are long, relative to those 300ms (mostly trials are longer than 5 seconds). Furthermore, these diagrams show that humans are always a bit slower than the ESEC algorithm.

Another interesting question about the comparison of the ESEC algorithm and human performance is that, if we map the moment of the prediction made by the individual in the action scenarios onto the related ESEC matrices, which column does it match? In spite of the slower human performance, are humans still using that same event column to perform prediction as the one used by the ESEC algorithm?

In fig.5.14, we pool responses across all people for each action. Column 0 (marked with the red arrow) in the histograms indicates that the human has recognized the action at the same ESEC column as the algorithm did, while column -1 and column +1 in the histograms show the human has predicted the action one column before or after the ESEC algorithm, respectively. We subtracted 300ms as pure sensor-motor reaction time before the analysis.

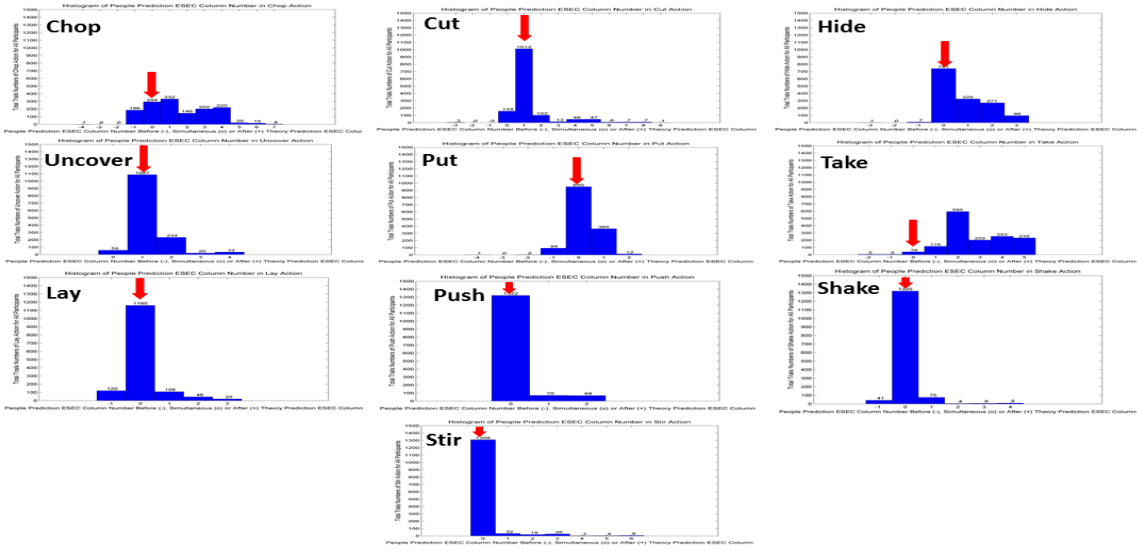


Figure 5.14: Histograms of the participants' median prediction event column number for all 10 manipulations. **Remarkably:** Column 0 gets the highest values for 6 of the 10 actions (Median!). Two actions (cut and uncover) are clearly recognized one column later. Chop is widely distributed but still with a Median of 1. Only Take is recognized quite a bit later.

For a more precise explanation, look at the enlarged sample of one of these actions in the fig.5.15. As we know, we have 10 actions in this experiment and 30 scenarios are available for each of them. On the other hand, there are 49 participants with an acceptable output. Thus, each individual observes 30 samples of a particular action, during the experiment and the action is observed in the total of **30*49=1470** times. The moments of the participants' predictions are mapped onto the ESEC

matrices and the prediction columns are obtained. Then we consider the median values of these column numbers for each action observation. According to the fig.5.15 in *Put on top* action, the median of the prediction columns in **950** of the total is in column **0**(the same column as the ESEC algorithm predicts there). Also, **365** and **12** of the total observations were happening in column **+1** column **+2**, hence shortly **after** column 0, while in **94** and **2** cases the median prediction columns are columns **-1** and **-2** (before the ESEC prediction column). There is also one case that predicts the type of action in column **-4** (4 columns before the ESEC prediction column). The total sum of these values is 1424, while the trials of each action are observed 1470 times. The difference is because in $1470 - 1424 = 46$ cases of the total, the *Put on top* action was not properly detected and we had only 1424 observations with the correct recognition. The rest of incorrect recognition are not included in the histogram.

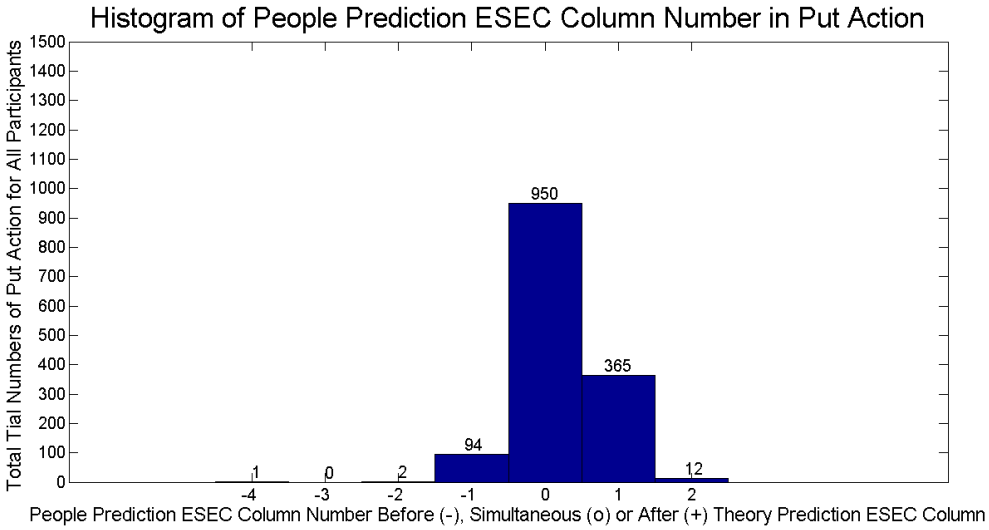


Figure 5.15: Histograms of the participants' median prediction event column number for the *Put on top* action

The general results in fig.5.14 is to some degree intriguing. It seems that for many actions humans "do it at the same time chunk" as the ESEC algorithm. *Chop*, *Take down* and *Hide* need a deeper look as they are for everyone rather more widely distributed.

Figures 5.16, 5.17 and 5.18 show the median of all participants' predictability power for each of 30 trials of these actions with standard deviation (STD). As can be seen, the variability of the predictability power is high in many trials, and this leads to the widespread distribution of the prediction column numbers in fig.5.14.

Cut and Uncover are also interesting and we should consider "how does Column +1 in the ESEC look like?" for these two actions. (Maybe Column +1 is highly indicative for this action here and Column 0 not so much).

To address this, the ESEC matrices of all these 10 manipulation actions and the event columns in which both, the ESEC predictions and the majority of human predictions happen, are explained in section 5.0.8.

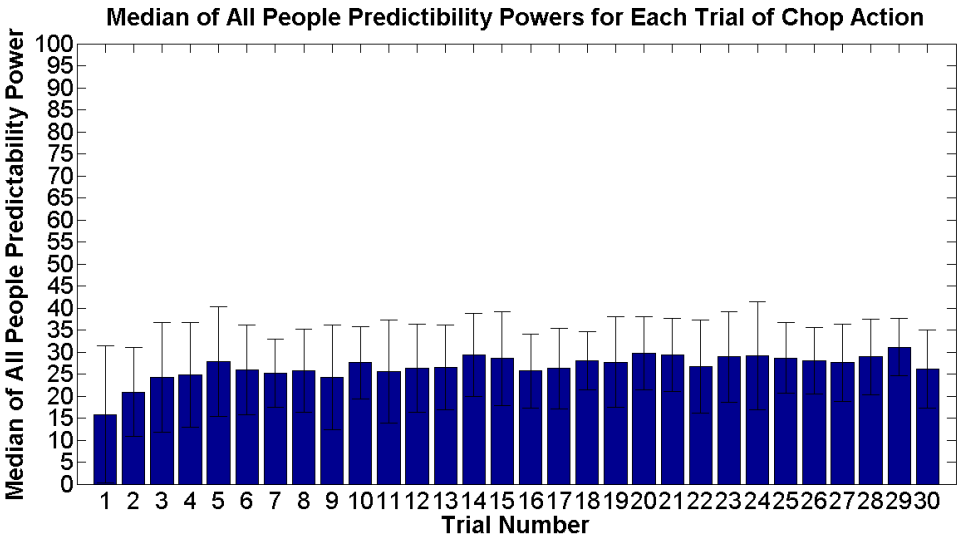


Figure 5.16: Median of all participants' predictability power for each trial of the *chop* action

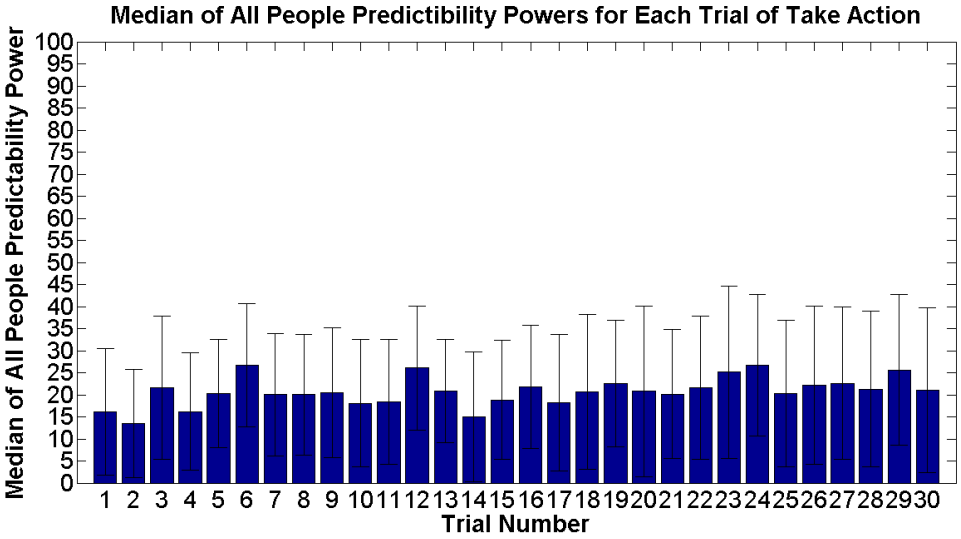


Figure 5.17: Median of all participants' predictability power for each trial of the *take* action

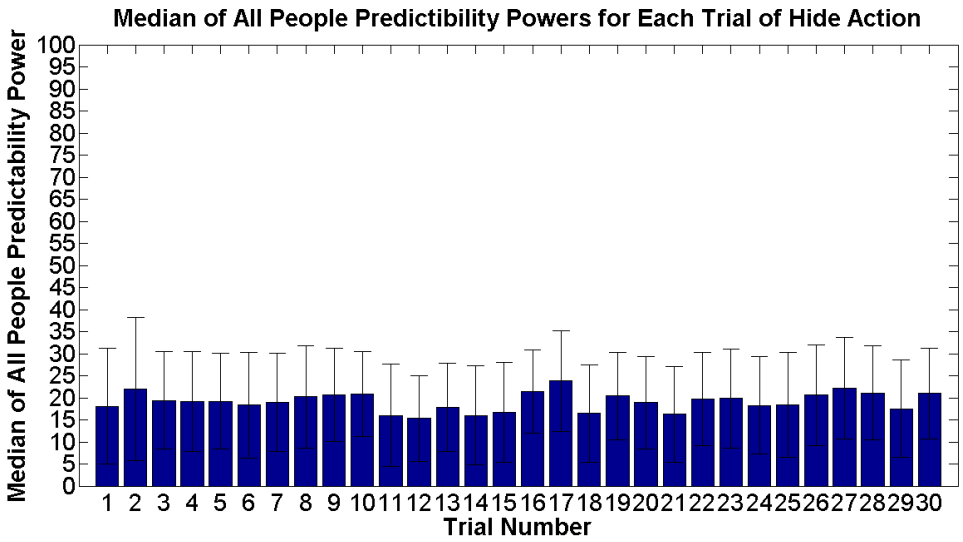


Figure 5.18: Median of all participants' predictability power for each trial of the *hide* action

5.0.8 ESEC Matrices

As a reminder, each ESEC matrix has 30 rows. The first 10 rows indicates the T/N relations between each pair of the fundamental objects (SEC matrix) and the middle and the last 10 rows indicate the static and dynamic spatial relations between those pairs, respectively. In this section, we need to show always the full version of our 10 manipulation actions matrices. The red marker at the bottom of each matrix marks the column, where the ESEC algorithm made the decision, while the violet marker at the top shows where the most of participants (according to the histograms of fig.5.0.8) have recognized this action. For each action:

1. First we denote the columns, which had been prediction columns for any of the people by light blue color following the distributions in figure 5.14.
2. Then we show all event transitions by a green color. For example if one spatial relation changes from ”MA” to ”HT” in two subsequent columns, we color both cells green. Exceptions:
   - Sometimes one relation changes continuously between the several columns (e.g.“MA”,“HT”,“MT”,... in columns i,i+1, i+2,...). Then we show this by a sequence of green color for that relation in the consecutive columns.
   - Sometimes a spatial relation changes in columns i and i+1 and also i+2 and i+3. In these case we show the first transition by light green and the second by dark green color. (e.g. the spatial relations between two items in 4 continuous columns could be: “Ar”,“ArT”,

"ArT","To", in this case the first transition (Ar$\rightarrow$ ArT in column i and i+1) is light green and the second transition ($ArT \rightarrow$ in column i+2 and i+3 is dark green.)

3. Also for better differentiation, we have spelt out the event of each blue column below the ESEC matrix in human terms.

1. **Chop**

H, 1	U	U	T	T	T	T	T	T	T	T	T	T	N	N
H, 2	U	U	U	U	N	N	N	N	X	X	X	X	X	X
H, 3	U	U	U	U	U	U	U	U	N	N	N	N	N	N
H, G	N	N	N	N	N	N	N	N	N	N	N	N	N	N
1, 2	U	U	U	U	T	T	T	T	X	X	X	X	X	X
1, 3	U	U	U	U	U	U	U	U	T	N	N	N	N	N
1, G	U	U	T	N	N	N	T	N	N	N	N	T	T	T
2, 3	U	U	U	U	U	U	U	U	X	X	X	X	X	X
2, G	U	U	U	U	T	T	T	T	X	X	X	X	X	X
3, G	U	U	U	U	U	U	U	U	T	T	T	T	T	T
H, 1	U	U	ArT	ArT	ArT	ArT	ArT	ArT	ArT	ArT	ArT	ArT	Ar	O
H, 2	U	U	U	U	Ab	Ab	Ab	Ab	X	X	X	X	X	X
H, 3	U	U	U	U	U	U	U	U	Ab	Ab	Ar	Ar	Ar	O
H, G	O	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	O
1, 2	U	U	U	U	To	in	in	To	X	X	X	X	X	X
1, 3	U	U	U	U	U	U	U	U	To	Ab	Ar	Ar	Ar	Ar
1, G	U	U	To	Ab	Ab	Ab	To	Ab	Ab	Ab	Ab	To	To	To
2, 3	U	U	U	U	U	U	U	U	X	X	X	X	X	X
2, G	U	U	U	U	To	To	To	To	X	X	X	X	X	X
3, G	U	U	U	U	U	U	U	U	To	To	To	To	To	To
H, 1	U	U	HT	MT	HT	MT	MT	MT	MT	MT	MT	MT	MA	Q
H, 2	U	U	U	U	S	S	S	S	X	X	X	X	X	X
H, 3	U	U	U	U	U	U	U	U	MA	MA	MA	S	MA	Q
H, G	Q	S	S	MA	S	S	S	S	S	S	GC	S	S	Q
1, 2	U	U	U	U	HT	FMT	FMT	FMT	X	X	X	X	X	X
1, 3	U	U	U	U	U	U	U	U	FMT	MA	MA	S	S	S
1, G	U	U	HT	MA	S	GC	HT	MA	MA	MA	GC	HT	HT	HT
2, 3	U	U	U	U	U	U	U	U	X	X	X	X	X	X
2, G	U	U	U	U	HT	HT	HT	HT	X	X	X	X	X	X
3, G	U	U	U	U	U	U	U	U	HT	HT	HT	HT	HT	HT
	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Figure 5.19: Chop ESEC Matrix

6) When object 1 starts to penetrate into the object 2. **7)** When the object 1 touches the ground. **8)** When object 1 exits from object 2. **9)** When object 1 is converted into two parts. **10)** When object 1 is moving apart from the new created objects. **11)** When the hand and object 1 are getting close to the ground. **12)** When the hand puts the object 1 on the ground. **13)** When the hand untouches object 1. **14)** When the hand is far away.

Fundamental Transition for ESEC Framework	Fundamental Transition for most of people
When the object 1 (e.g. Knife) touches the ground (e.g. bottom of the fruit).	When the object 1 (e.g. Knife) leaves the inside of the object 2 and getting distance from the ground (e.g. bottom of the fruit).

2. **Cut**

H, 1	U	U	T	T	T	T	T	T	T	T	T	N	N
H, 2	U	U	U	U	N	N	N	X	X	X	X	X	X
H, 3	U	U	U	U	U	U	U	N	N	N	N	N	N
H, G	N	N	N	N	N	N	N	N	N	N	N	N	N
1, 2	U	U	U	U	T	T	T	X	X	X	X	X	X
1, 3	U	U	U	U	U	U	U	T	N	N	N	N	N
1, G	U	U	T	N	N	N	N	N	N	N	T	T	T
2, 3	U	U	U	U	U	U	U	X	X	X	X	X	X
2, G	U	U	U	U	T	T	T	X	X	X	X	X	X
3, G	U	U	U	U	U	U	U	T	T	T	T	T	T
H, 1	U	U	ArT	ArT	ArT	ArT	ArT	ArT	ArT	ArT	ArT	Ar	O
H, 2	U	U	U	U	Ab	Ab	Ab	X	X	X	X	X	X
H, 3	U	U	U	U	U	U	U	Ab	Ab	Ar	Ar	Ar	O
H, G	O	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	O
1, 2	U	U	U	U	To	in	To	X	X	X	X	X	X
1, 3	U	U	U	U	U	U	U	To	Ab	Ar	Ar	Ar	Ar
1, G	U	U	To	Ab	Ab	Ab	Ab	Ab	Ab	Ab	To	To	To
2, 3	U	U	U	U	U	U	U	X	X	X	X	X	X
2, G	U	U	U	U	To	To	To	X	X	X	X	X	X
3, G	U	U	U	U	U	U	U	To	To	To	To	To	To
H, 1	U	U	HT	MT	MT	MT	MT	MT	MT	MT	MT	MA	Q
H, 2	U	U	U	U	S	S	S	X	X	X	X	X	X
H, 3	U	U	U	U	U	U	U	MA	MA	MA	S	MA	Q
H, G	Q	S	S	MA	S	S	S	S	S	GC	S	S	Q
1, 2	U	U	U	U	FMT	FMT	FMT	X	X	X	X	X	X
1, 3	U	U	U	U	U	U	U	FMT	MA	MA	S	S	S
1, G	U	U	HT	MA	S	GC	MA	MA	MA	GC	HT	HT	HT
2, 3	U	U	U	U	U	U	U	X	X	X	X	X	X
2, G	U	U	U	U	HT	HT	HT	X	X	X	X	X	X
3, G	U	U	U	U	U	U	U	HT	HT	HT	HT	HT	HT
	1	2	3	4	5	6	7	8	9	10	11	12	13

Figure 5.20: Cut ESEC Matrix

2) When the hand moves above the ground. **5)** When object 1 starts moving on the surface of object 2. **6)** When object 1 starts to penetrate into object 2. **7)** When object 1 starts to leave object 2. **8)** When object 1 is divided to two parts. **9)** When object 1 is moving apart from the new created objects. **10)** When the hand and object 1 are getting close to the ground. **11)** When the hand puts object 1 on the ground. **12)** When the hand untouches object 1. **13)** When the hand is far away.

Fundamental Transition for ESEC Framework	Fundamental Transition for most of people
When the object 1 (e.g. Knife) starts moving on the surface object 2 (e.g. fruit).	When the object 1 (e.g. Knife) goes inside of the object 2 (e.g. fruit).

3. **Hide**

H, 1	U	U	T	T	T	T	N	N
H, 2	U	U	U	U	N	A	A	A
H, 3	U	U	U	U	U	U	U	U
H, G	N	N	N	N	N	N	N	N
1, 2	U	U	U	U	T	A	A	A
1, 3	U	U	U	U	U	U	U	U
1, G	U	U	T	N	N	T	T	T
2, 3	U	U	U	U	U	U	U	U
2, G	U	U	U	U	T	A	A	A
3, G	U	U	U	U	U	U	U	U
H, 1	U	U	ArT	ArT	ArT	ArT	Ab	O
H, 2	U	U	U	U	Ab	A	A	A
H, 3	U	U	U	U	U	U	U	U
H, G	O	Ab	Ab	Ab	Ab	Ab	Ar	O
1, 2	U	U	U	U	To	A	A	A
1, 3	U	U	U	U	U	U	U	U
1, G	U	U	To	Ab	Ab	To	To	To
2, 3	U	U	U	U	U	U	U	U
2, G	U	U	U	U	To	A	A	A
3, G	U	U	U	U	U	U	U	U
H, 1	U	U	HT	MT	MT	HT	MA	Q
H, 2	U	U	U	U	GC	A	A	A
H, 3	U	U	U	U	U	U	U	U
H, G	Q	S	S	MA	GC	S	S	Q
1, 2	U	U	U	U	FMT	A	A	A
1, 3	U	U	U	U	U	U	U	U
1, G	U	U	HT	MA	GC	HT	HT	HT
2, 3	U	U	U	U	U	U	U	U
2, G	U	U	U	U	HT	A	A	A
3, G	U	U	U	U	U	U	U	U
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Figure 5.21: Hide ESEC Matrix

4) When the hand and object 1 are moving away from the ground. **5)** When object 1 starts covering object 2. **6)** When object 1 completely covers object 2 (object 2 is absent). **7)** When the hand untouches object 1. **8)** When the hand is far away.

Fundamental Transition for ESEC Framework	**Fundamental Transition for most of people**
When the object 1 (upper object) places on the top of the object 2 (lower object) and at a same time getting close to the ground while it (object 1) is moving along of the object 2.	When the object 1 (upper object) places on the top of the object 2 (lower object) and at a same time getting close to the ground while it (object 1) is moving along of the object 2.

4. **Uncover**

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

Figure 5.22: Uncover ESEC Matrix

4) When object 1 starts moving away from the ground and the covered object begins to appear. **5)** When object 1 untouches object 2. **6)** When the hand and object 1 are getting close to the ground. **7)** When the hand puts object 1 on the ground.

Fundamental Transition for ESEC Framework	Fundamental Transition for most of people
When the object 1 (upper object) touches top of the object 2 and at a same time moving apart from the ground while it (object 1) is moving along of the object 2.	When the object 1 (cover object) untouches the object 2 (hided object) totally and moving apart from it.

5. **Put**

H, 1	U	U	T	T	T	N	N	A
H, 2	U	U	U	U	N	N	N	A
H, 3	U	U	U	U	U	U	U	U
H, G	N	N	N	N	N	N	N	A
1, 2	U	U	U	U	T	T	T	T
1, 3	U	U	U	U	U	U	U	U
1, G	U	U	T	N	N	N	N	N
2, 3	U	U	U	U	U	U	U	U
2, G	U	U	U	U	T	T	T	T
3, G	U	U	U	U	U	U	U	U
H, 1	U	U	ArT	ArT	ArT	Ar	O	A
H, 2	U	U	U	U	Ab	Ab	O	A
H, 3	U	U	U	U	U	U	U	U
H, G	O	Ab	Ab	Ab	Ab	Ab	O	A
1, 2	U	U	U	U	To	To	To	To
1, 3	U	U	U	U	U	U	U	U
1, G	U	U	To	Ab	Ab	Ab	Ab	Ab
2, 3	U	U	U	U	U	U	U	U
2, G	U	U	U	U	To	To	To	To
3, G	U	U	U	U	U	U	U	U
H, 1	U	U	HT	MT	HT	MA	Q	A
H, 2	U	U	U	U	S	MA	Q	A
H, 3	U	U	U	U	U	U	U	U
H, G	Q	S	S	MA	S	S	Q	A
1, 2	U	U	U	U	HT	HT	HT	HT
1, 3	U	U	U	U	U	U	U	U
1, G	U	U	HT	MA	S	S	S	S
2, 3	U	U	U	U	U	U	U	U
2, G	U	U	U	U	HT	HT	HT	HT
3, G	U	U	U	U	U	U	U	U
	1	2	3	4	5	6	7	8

Figure 5.23: Put ESEC Matrix

5) When the hand puts object 2 on object 1. **6)** When the hand untouches object 2. **7)** When the hand is far away. **8)** When the hand leaves the scene.

Fundamental Transition for ESEC Framework	Fundamental Transition for most of people
When the hand leaves the upper object.	When the hand leaves the upper object.

6. **Take**

H, 1	U	U	T	T	T	T	N	N
H, 2	U	U	N	N	N	N	N	N
H, 3	U	U	U	U	U	U	U	U
H, G	N	N	N	N	N	N	N	N
1, 2	U	U	T	N	N	N	N	N
1, 3	U	U	U	U	U	U	U	U
1, G	U	U	N	N	N	T	T	T
2, 3	U	U	U	U	U	U	U	U
2, G	U	U	T	T	T	T	T	T
3, G	U	U	U	U	U	U	U	U
H, 1	U	U	ArT	ArT	ArT	ArT	Ar	O
H, 2	U	U	Ab	Ab	Ar	Ar	Ar	O
H, 3	U	U	U	U	U	U	U	U
H, G	O	Ab	Ab	Ab	Ab	Ab	Ab	O
1, 2	U	U	To	Ab	Ab	Ar	Ar	Ar
1, 3	U	U	U	U	U	U	U	U
1, G	U	U	Ab	Ab	Ab	To	To	To
2, 3	U	U	U	U	U	U	U	U
2, G	U	U	To	To	To	To	To	To
3, G	U	U	U	U	U	U	U	U
H, 1	U	U	HT	MT	MT	HT	MA	Q
H, 2	U	U	S	MA	MA	S	MA	Q
H, 3	U	U	U	U	U	U	U	U
H, G	Q	S	S	S	GC	S	S	Q
1, 2	U	U	HT	MA	MA	S	S	S
1, 3	U	U	U	U	U	U	U	U
1, G	U	U	S	MA	GC	HT	HT	HT
2, 3	U	U	U	U	U	U	U	U
2, G	U	U	HT	HT	HT	HT	HT	HT
3, G	U	U	U	U	U	U	U	U
	1	2	3	4	5	6	7	8

Figure 5.24: Take ESEC Matrix

1) When the hand is far away. **2)** When the hand moves above the ground. **3)** When the hand touches object 1. **4)** When object 1 is separated from object 2 by the hand. **5)** When the hand and object 1 are getting close to the ground. **6)** When the hand puts object 1 on the ground. **7)** When the hand untouches object 1. **8)** When the hand is far away.

Fundamental Transition for ESEC Framework	Fundamental Transition for most of people
When the hand touches the object 1 while the object 1 (itself) is on the top of object 2.	When the hand and the touched object (after removing the object 1 from the object 2) are getting close to the ground.

7. **Lay**

H, 1	U	U	T	T	T	N	N
H, 2	U	U	U	U	U	U	U
H, 3	U	U	U	U	U	U	U
H, G	N	N	N	N	N	N	N
1, 2	U	U	U	U	U	U	U
1, 3	U	U	U	U	U	U	U
1, G	U	U	T	T	T	T	T
2, 3	U	U	U	U	U	U	U
2, G	U	U	U	U	U	U	U
3, G	U	U	U	U	U	U	U
H, 1	U	U	ArT	To	To	Ab	O
H, 2	U	U	U	U	U	U	U
H, 3	U	U	U	U	U	U	U
H, G	O	Ab	Ab	Ab	Ab	Ab	O
1, 2	U	U	U	U	U	U	U
1, 3	U	U	U	U	U	U	U
1, G	U	U	To	To	To	To	To
2, 3	U	U	U	U	U	U	U
2, G	U	U	U	U	U	U	U
3, G	U	U	U	U	U	U	U
H, 1	U	U	HT	MT	HT	MA	Q
H, 2	U	U	U	U	U	U	U
H, 3	U	U	U	U	U	U	U
H, G	Q	S	S	S	S	S	Q
1, 2	U	U	U	U	U	U	U
1, 3	U	U	U	U	U	U	U
1, G	U	U	HT	FMT	HT	HT	HT
2, 3	U	U	U	U	U	U	U
2, G	U	U	U	U	U	U	U
3, G	U	U	U	U	U	U	U
	1	2	3	4	5	6	7

Figure 5.25: Lay ESEC Matrix

3) When the hand touches object 1 on the ground. **4)** When the hand starts laying object 1 on the ground. **5)** When the hand stops moving object 1. **6)** When the hand untouches object 1. **7)** When the hand is far away.

Fundamental Transition for ESEC Framework	Fundamental Transition for most of people
When the object 1 starts moving with the hand on the ground while their static relation is changing during the movement.	When the object 1 starts moving with the hand on the ground while their static relation is changing during the movement.

8. **Push**

H, 1	U	U	T	T	N	N
H, 2	U	U	U	U	U	U
H, 3	U	U	U	U	U	U
H, G	N	N	N	N	N	N
1, 2	U	U	U	U	U	U
1, 3	U	U	U	U	U	U
1, G	U	U	T	T	T	T
2, 3	U	U	U	U	U	U
2, G	U	U	U	U	U	U
3, G	U	U	U	U	U	U
H, 1	U	U	ArT	ArT	Ar	O
H, 2	U	U	U	U	U	U
H, 3	U	U	U	U	U	U
H, G	O	Ab	Ab	Ab	Ab	O
1, 2	U	U	U	U	U	U
1, 3	U	U	U	U	U	U
1, G	U	U	To	To	To	To
2, 3	U	U	U	U	U	U
2, G	U	U	U	U	U	U
3, G	U	U	U	U	U	U
H, 1	U	U	HT	MT	MA	Q
H, 2	U	U	U	U	U	U
H, 3	U	U	U	U	U	U
H, G	Q	S	S	S	S	Q
1, 2	U	U	U	U	U	U
1, 3	U	U	U	U	U	U
1, G	U	U	HT	FMT	HT	HT
2, 3	U	U	U	U	U	U
2, G	U	U	U	U	U	U
3, G	U	U	U	U	U	U
	1	2	3	4	5	6

Figure 5.26: Push ESEC Matrix

4) When the hand starts pushing object 1. **5)** When the hand untouches object 1. **6)** When the hand is far away.

Fundamental Transition for ESEC Framework	Fundamental Transition for most of people
When the hand starts moving the object.	When the hand starts moving the object.

9. **Shake**

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

Figure 5.27: Shake ESEC Matrix

4) When the hand separates object 1 from the ground and lifts it. **5)** When the hand starts shaking object 1. **6)** When the hand and object 1 are getting close to the ground. **7)** When the hand puts object 1 on the ground. **8)** When the hand untouches object 1. **9)** When the hand is far away.

Fundamental Transition for ESEC Framework	Fundamental Transition for most of people
When the object (and also hand) remains at a steady distance from the ground while still is moving with hand (MT).	When the object (and also hand) remains at a steady distance from the ground while still is moving with hand (MT).

10. **Stir**

H, 1	U	U	T	T	T	T	T	T	T	T	T	T	N	N
H, 2	U	U	U	U	N	N	N	N	N	N	N	N	N	N
H, 3	U	U	U	U	U	U	U	U	U	U	U	U	U	U
H, G	N	N	N	N	N	N	N	N	N	N	N	N	N	N
1, 2	U	U	U	U	T	T	T	T	T	N	N	N	N	N
1, 3	U	U	U	U	U	U	U	U	U	U	U	U	U	U
1, G	U	U	T	N	N	N	N	N	N	N	N	T	T	T
2, 3	U	U	U	U	U	U	U	U	U	U	U	U	U	U
2, G	U	U	U	U	T	T	T	T	T	T	T	T	T	T
3, G	U	U	U	U	U	U	U	U	U	U	U	U	U	U
H, 1	U	U	ArT	ArT	ArT	ArT	ArT	ArT	ArT	ArT	ArT	ArT	Ar	O
H, 2	U	U	U	U	Ab	Ab	Ab	Ab	Ab	Ab	Ar	Ar	Ar	O
H, 3	U	U	U	U	U	U	U	U	U	U	U	U	U	U
H, G	O	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	O
1, 2	U	U	U	U	To	in	in	in	To	Ab	Ar	Ar	Ar	Ar
1, 3	U	U	U	U	U	U	U	U	U	U	U	U	U	U
1, G	U	U	To	Ab	Ab	Ab	Ab	Ab	Ab	Ab	Ab	To	To	To
2, 3	U	U	U	U	U	U	U	U	U	U	U	U	U	U
2, G	U	U	U	U	To	To	To	To	To	To	To	To	To	To
3, G	U	U	U	U	U	U	U	U	U	U	U	U	U	U
H, 1	U	U	HT	MT	HT	MT	MT	MT	HT	MT	MT	HT	MA	Q
H, 2	U	U	U	U	S	S	S	S	S	MA	MA	S	MA	Q
H, 3	U	U	U	U	U	U	U	U	U	U	U	U	U	U
H, G	Q	S	S	MA	S	S	S	S	S	S	S	S	S	Q
1, 2	U	U	U	U	HT	FMT	FMT	FMT	HT	MA	MA	S	S	S
1, 3	U	U	U	U	U	U	U	U	U	U	U	U	U	U
1, G	U	U	HT	MA	S	GC	S	MA	S	S	GC	HT	HT	HT
2, 3	U	U	U	U	U	U	U	U	U	U	U	U	U	U
2, G	U	U	U	U	HT	HT	HT	HT	HT	HT	HT	HT	HT	HT
3, G	U	U	U	U	U	U	U	U	U	U	U	U	U	U
	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Figure 5.28: Stir ESEC Matrix

7) When object 1 starts stirring object 2. **8)** When object 1 starts leaving object 2. **9)** When object 2 is exactly on top of object 1. **10)** When object 1 untouches object 2. **11)** When the hand and object 1 are getting close to the ground. **12)** When the hand puts object 1 on the ground. **13)** When the hand untouches object 1.

Fundamental Transition for ESEC Framework	Fundamental Transition for most of people
When the object 1 (e.g. spoon) goes inside of the object 2 (liquid container), placed at a certain distance from the bottom of the liquid container and starts to spin.	When the object 1 (e.g. spoon) goes inside of the object 2 (liquid container), placed at a certain distance from the bottom of the liquid container and starts to spin.

Chapter 6

Summary and Future Remarks

This chapter concludes this thesis with a short summary and final remarks. All previous chapters, explained so far, include their own summary sections in which main findings with corresponding advantages and drawbacks are discussed. In this chapter, we will first highlight the most important points of each chapter and then continue with the problems of ESEC framework and future remarks.

6.0.1 Summary

In this thesis, we introduced a framework for semantic representation of manipulation actions, named as "Enriched Semantic Event Chain (ESEC)", which focuses on spatial relations between objects of a scene. We divided possible spatial relations into "static" and "dynamic" relations. ESEC creates a temporal sequence of static and dynamic spatial relations between the objects that take part in the manipulation aiding early action recognition. Mathematically speaking, ESECs are transition matrices that symbolically encode the relational static and dynamic changes between (unspecified) objects. Each row of an ESEC matrix represents the sequence of the spatial relations between each pair of manipulated objects attained during the continuous video. Whenever a change occurs in any of those spatial relations a new column is created. As a consequence, every column reflects at least one such change. In order to facilitate the spatial relations computations, we model each object in a simple AABB (Axis-Aligned Bounding Box) and perform calculations based on the relationships between the AABBs. Accordingly, we suggested a method for recognition of manipulations and tested it on MANIAC data-set.

Moreover, we proposed an approach for the prediction of manipulation actions, based on the ESEC framework and compared it with the original SEC and an "object-free" Hidden Markov Model (HMM)- based method. We showed that on average, the ESEC framework outperforms both SEC and HMM-based methods. One possible strength of ESEC is that it does not rely

on time-continuous information, making it considerably less prone to variability (and noise), compared to when using the *quasi-symbolic* representations. Indeed, when watching some of the examples in the MANIAC data set, it is perceivable that time continuous information does not necessarily improve prediction much, because the only aspect added by this type of information is the action dynamics. Dynamics do not influence the action class but will play a role in how an action is executed (e.g. fast versus slow, etc.)

In the next step, we compared our method's performance in predicting manipulation actions with that of humans, by selecting 10 actions which are distributed in all possible groups and subgroups of manipulations and constructing 300 scenarios of those actions. We provided an experiment in the virtual reality environment and compared the predictability power of the ESEC algorithm with 50 human participants and analyzed the results. We finally came to the conclusion that the present algorithm works better than humans, as well as a mathematical HMM based method.

ESEC framework does not require any object recognition, action trajectories, shape features or action reconstruction and performs the recognition tasks only by using semantic representation and spatial relations in a simple way which makes it unique as of other studies. Furthermore, we designed a noise reduction pre-processing algorithm which eliminates those errors occurring due to the presence of noise. ESEC has a negligible complexity, can perform in real time scenarios and is strongly linked to the way human language describes an action. Several psychological investigations have reported that event-based en- coding might be fundamental to action understanding in the brain (see e.g. [57]) and, thus, our framework might indeed be better linked to the way humans understand actions.

6.0.2 Problems of ESEC Framework

- Our framework heavily relies on the segment permanence (i.e., reliable tracking) which is performed by advanced computer vision methods and we are aware that failures in the computer vision can harm our approach. Clearly, on the computer vision side, improvements can be made to better assure this, which is not in the core of this thesis.
- We modeled each object in an Axis Aligned Bounding Box (AABB) which is not an accurate model for objects with concave shapes. AABB model was selected because of its low complexity and simplicity of calculation allowing ESEC to run online. Although we have already achieved improved results at the moment, we can get more accurate results using the oriented bounding box (OBB) for the object modeling.

6.0.3 Future Remarks

According to the insight gained in this work, the following paths are suggested for future works:

- Currently, ESEC framework is used in representation, discrimination, recognition and prediction of single hand manipulation actions (or two hands in the cases that one hand only supports an object), we are nevertheless planning to extend this approach to dual arms manipulations or interactions of two hands. In this regards, the first step is to extract Dynamic Motion Primitives (DMPs) which are used for robot executions and the second step is to define the timing relationships between DMPs according to a Context Free Grammar (CFG) based structure. The extension of the current ontology to bi-manual tasks enables us to perform more/better actions while introducing new challenges.
- The key feature of our suggested framework is that it does not rely on any object recognition, but that it equips ESEC with additional information such as affordances of objects potential of having a significant effect on the prediction speed.
- Given the tight link between Natural Language Processing (NLP) and scene description, we can incorporate Natural Language Processing (NLP) and Learning from Demonstration (LfD) into our framework for future works to further enhance the performance of our approach.

Bibliography

[1] L. M. Ma, T. Fong, M. J. Micire, Y. K. Kim, and K. Feigh, "Human-robot teaming: concepts and components for design," in *Field and Service Robotics*. Springer, 2018, pp. 649–663.

[2] J. Wirtz, P. G. Patterson, W. H. Kunz, T. Gruber, V. N. Lu, S. Paluch, and A. Martins, "Brave new world: service robots in the frontline," *Journal of Service Management*, vol. 29, no. 5, pp. 907–931, 2018.

[3] B. Graf, M. Hans, and R. D. Schraft, "Care-o-bot iidevelopment of a next generation robotic home assistant," *Autonomous robots*, vol. 16, no. 2, pp. 193–205, 2004.

[4] M. Alač, J. Movellan, and F. Tanaka, "When a robot is social: Spatial arrangements and multimodal semiotic engagement in the practice of social robotics," *Social Studies of Science*, vol. 41, no. 6, pp. 893–926, 2011.

[5] I. Bloch, "Fuzzy spatial relationships for image processing and interpretation: a review," *Image and Vision Computing*, vol. 23, no. 2, pp. 89–110, 2005.

[6] S. Aksoy, C. Tusk, K. Koperski, and G. Marchisio, "Scene modeling and image mining with a visual grammar," in *Frontiers of remote sensing information processing*. World Scientific, 2003, pp. 35–62.

[7] M. J. Aein, "Development and analysis of a library of actions for robot arm-hand systems," Ph.D. dissertation, Georg-August-Universität Göttingen, 2016.

[8] C. Fermüller, F. Wang, Y. Yang, K. Zampogiannis, Y. Zhang, F. Barranco, and M. Pfeiffer, "Prediction of manipulation actions," *International Journal of Computer Vision*, vol. 126, no. 2-4, pp. 358–374, 2018.

[9] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "Hand gesture recognition based on combined features extraction," *World Academy of Science, Engineering and Technology*, vol. 60, p. 395, 2009.

[10] J. Singha and R. H. Laskar, "Hand gesture recognition using two-level speed normalization, feature selection and classifier fusion," *Multimedia Systems*, vol. 23, no. 4, pp. 499–514, 2017.

[11] I. Bloch and A. Ralescu, "Directional relative position between objects in image processing: a comparison between fuzzy approaches," *pattern Recognition*, vol. 36, no. 7, pp. 1563–1582, 2003.

[12] O. Colliot, O. Camara, and I. Bloch, "Integration of fuzzy spatial relations in deformable modelsapplication to brain mri segmentation," *Pattern recognition*, vol. 39, no. 8, pp. 1401–1414, 2006.

[13] J. M. Keller and X. Wang, "A fuzzy rule-based approach to scene description involving spatial relationships," *Computer Vision and Image Understanding*, vol. 80, no. 1, pp. 21–41, 2000.

[14] M. Bhatt, F. Dylla, and J. Hois, "Spatio-terminological inference for the design of ambient environments," in *International Conference on Spatial Information Theory*. Springer, 2009, pp. 371–391.

[15] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[16] K. Sjöö, A. Aydemir, and P. Jensfelt, "Topological spatial relations for active visual search," *Robotics and Autonomous Systems*, vol. 60, no. 9, pp. 1093–1107, 2012.

[17] S. S. Ge, *Autonomous mobile robots: sensing, control, decision making and applications*. CRC press, 2006.

[18] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[19] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 5. IEEE, 2004, pp. 4845–4851.

[20] R. Moratz and T. Tenbrink, "Spatial reference in linguistic human-robot interaction: Iterative, empirically supported development of a model of projective relations," *Spatial cognition and computation*, vol. 6, no. 1, pp. 63–107, 2006.

[21] C. Landsiedel, V. Rieser, M. Walter, and D. Wollherr, "A review of spatial reasoning and interaction for real-world robotics," *Advanced Robotics*, vol. 31, no. 5, pp. 222–242, 2017.

[22] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1398–1403.

[23] R. Dillmann, T. Asfour, M. Do, R. Jäkel, A. Kasper, P. Azad, A. Ude, S. R. Schmidt-Rohr, and M. Lösch, "Advances in robot programming by demonstration," *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 295–303, 2010.

[24] A. Ude, "Trajectory generation from noisy positions of object features for teaching robot paths," *Robotics and Autonomous Systems*, vol. 11, no. 2, pp. 113–127, 1993.

[25] D. Lee and Y. Nakamura, "Stochastic model of imitating a new observed motion based on the acquired motion primitives," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 4994–5000.

[26] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.

[27] T. Luksch, M. Gienger, M. Mühlig, and T. Yoshiike, "A dynamical systems approach to adaptive sequencing of movement primitives," in *ROBOTIK 2012; 7th German Conference on Robotics*. VDE, 2012, pp. 1–6.

[28] M. Sridhar, A. G. Cohn, and D. C. Hogg, "Learning functional object categories from a relational spatio-temporal representation," in *ECAI 2008: 18th European Conference on Artificial Intelligence (Frontiers in Artificial Intelligence and Applications)*. IOS Press, 2008, pp. 606–610.

[29] 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.

[30] K. Ramirez-Amaro, E.-S. Kim, J. Kim, B.-T. Zhang, M. Beetz, and G. Cheng, "Enhancing human action recognition through spatio-temporal feature learning and semantic rules," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2013, pp. 456–461.

[31] K. Ramirez-Amaro, M. Beetz, and G. Cheng, "Extracting semantic rules from human observations," in *ICRA workshop: Semantics, Identification and Control of Robot-Human-Environment Interaction*, 2013.

[32] E. E. Aksoy, A. Orhan, and F. Wörgötter, "Semantic decomposition and recognition of long and complex manipulation action sequences," *International Journal of Computer Vision*, vol. 122, no. 1, pp. 84–115, 2017.

[33] K. Nagahama, K. Yamazaki, K. Okada, and M. Inaba, "Manipulation of multiple objects in close proximity based on visual hierarchical relationships," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1303–1310.

[34] Y. Yang, C. Fermuller, and Y. Aloimonos, "A cognitive system for human manipulation action understanding," in *the Second Annual Conference on Advances in Cognitive Systems (ACS)*, vol. 2. Citeseer, 2013.

[35] D. R. Faria, R. Martins, J. Lobo, and J. Dias, "Extracting data from human manipulation of objects towards improving autonomous robotic grasping," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 396–410, 2012.

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

[37] K. Zampogiannis, Y. Yang, C. Fermüller, and Y. Aloimonos, "Learning the spatial semantics of manipulation actions through preposition grounding," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 1389–1396.

[38] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "Hand trajectory-based gesture spotting and recognition using hmm," in *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 3577–3580.

[39] F. Ziaeetabar, T. Kulvicius, M. Tamosiunaite, and F. Wörgötter, "Recognition and prediction of manipulation actions using enriched semantic event chains," *Robotics and Autonomous Systems*, vol. 110, pp. 173–188, 2018.

[40] 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.

[41] 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.

[42] M. Robiony, I. Salvo, F. Costa, N. Zerman, M. Bazzocchi, F. Toso, C. Bandera, S. Filippi, M. Felice, and M. Politi, "Virtual reality surgical planning for maxillofacial distraction osteogenesis: the role of reverse engineering rapid prototyping and cooperative work," *Journal of oral and maxillofacial surgery*, vol. 65, no. 6, pp. 1198–1208, 2007.

[43] S. K. Ong and A. Y. C. Nee, *Virtual and augmented reality applications in manufacturing*. Springer Science & Business Media, 2013.

[44] A. Van Dam, A. S. Forsberg, D. H. Laidlaw, J. J. LaViola, and R. M. Simpson, "Immersive vr for scientific visualization: A progress report," *IEEE Computer Graphics and Applications*, vol. 20, no. 6, pp. 26–52, 2000.

[45] R. S. Kalawsky, "The science of virtual reality and virtual environments: a technical, scientific and engineering reference on virtual environments," 1996.

[46] Z. Merchant, E. T. Goetz, L. Cifuentes, W. Keeney-Kennicutt, and T. J. Davis, "Effectiveness of virtual reality-based instruction on students' learning outcomes in k-12 and higher education: A meta-analysis," *Computers & Education*, vol. 70, pp. 29–40, 2014.

[47] W. S. Kim, "Computer vision assisted virtual reality calibration," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 450–464, 1999.

[48] N. Ayache, "Medical computer vision, virtual reality and robotics," *Image and Vision Computing*, vol. 13, no. 4, pp. 295–313, 1995.

[49] F. Ziaeetabar, T. Kulvicius, M. Tamosiunaite, and F. Wörgötter, "Prediction of manipulation action classes using semantic spatial reasoning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3350–3357.

[50] W. Winn, "A conceptual basis for educational applications of virtual reality," *Technical Publication R-93-9, Human Interface Technology Laboratory of the Washington Technology Center, Seattle: University of Washington*, 1993.

[51] S. Zhang, J. Teizer, J.-K. Lee, C. M. Eastman, and M. Venugopal, "Building information modeling (bim) and safety: Automatic safety checking of construction models and schedules," *Automation in Construction*, vol. 29, pp. 183–195, 2013.

[52] T. S. Mujber, T. Szecsi, and M. S. Hashmi, "Virtual reality applications in manufacturing process simulation," *Journal of materials processing technology*, vol. 155, pp. 1834–1838, 2004.

[53] M. O. Onyesolu and F. U. Eze, "Understanding virtual reality technology: advances and applications," in *Advances in Computer Science and Engineering*. IntechOpen, 2011.

[54] S. Pfeiffer, "Virtual reality system for action prediction," Bachelor Thesis, Göttingen University, January 2019.

[55] A. V. Zaitsev and Y. A. Skorik, "Mathematical description of sensorimotor reaction time distribution," *Human Physiology*, vol. 28, no. 4, pp. 494–497, 2002.

[56] C. Cabib, S. Llufriu, J. Casanova-Molla, A. Saiz, and J. Valls-Solé, "Defective sensorimotor integration in preparation for reaction time tasks in patients with multiple sclerosis," *American Journal of Physiology-Heart and Circulatory Physiology*, 2014.

[57] B. Hommel, "The theory of event coding (tec) as embodied-cognition framework," *Frontiers in Psychology*, vol. 6, pp. 1–5, 2015.