Proceedings of the Second Annual Conference on Advances in Cognitive Systems ACS-2013 (109-124)

A Cognitive System for Human Manipulation Action Understanding

Yezhou Yang YZYANG@CS.UMD.EDU
Cornelia Fermüller FER@UMIACS.UMD.EDU
Yiannis Aloimonos YIANNIS@CS.UMD.EDU
University of Maryland Institute for Advanced Computer Studies, College Park, MD, 20740, USA

Abstract

This paper describes the architecture of a cognitive system that interprets human manipulation actions from perceptual information (image and depth data) and consists of perceptual modules and reasoning modules that are in interaction with each other. The contributions of our work are along two core problems at the heart of action understanding: a.) the grounding of relevant information about actions in perception (the perception - action integration problem), and b.) the organization of perceptual and high-level symbolic information for interpreting the actions (the sequencing problem). At the high level, actions are represented with the Manipulation Action Grammar, a context-free grammar and associated parsing algorithms, which organizes actions as a sequence of sub-events. Each sub-event is described by the hand, movements and the objects and tools involved, and the relevant information about these quantities is obtained from biological-inspired perception modules. These modules track the hands and objects and recognize the hand grasp, objects and actions using attention, segmentation, and feature description. Experiments on a new dataset of manipulation actions show that our system can successfully extract the relevant visual information and semantic representation. This representation could further be used by the cognitive agent for reasoning, prediction, and planning.

1. Introduction

Cognitive systems interacting with humans need to be able to interpret human actions. Here we are concerned with manipulation actions, that is actions performed by agents (humans or robots) on objects, which result in some physical change of the object. There has been much work recently on action recognition, with most studies considering short lived actions, where the beginning and end of the sequence is defined. Most efforts then were devoted to the recognition of movements and associated objects, both problems of great interest to the study of perception. However, the more complex an action becomes, the less reliable individual perceptual events are for the characterization of actions. Thus, the problem of interpreting manipulation actions, because of the large variation in which humans can perform these actions, involves many more challenges than simple recognizing movements and objects.

Since perceptual events do not suffice, then how do we determine the beginning and end of action segments, and how do we combine the individual segments into longer segments corresponding to a manipulation action? An essential component in the description of manipulations is the

underlying goal. The goal of a manipulation action is the physical change induced on the object. To accomplish it the hands need to perform a sequence of sub-actions on the object, and a subaction is when the hand grasps or releases the object, or the hand changes grasp type during a movement.. Centered around this idea, we developed a grammatical formalism for parsing and interpreting action sequences, and developed the vision modules to obtain from videos the symbolic information used in the grammatical structure.

Our formalism for describing manipulation actions uses a structure similar to natural language. What do we gain from a formal description of action, though? This is equal to asking what one gains from a formal description of language as a generative system. Chomsky's contribution to language research, was exactly this: the formal description of language through the formulation of the Generative and Transformational Grammar (Chomsky, 1957). It revolutionized language research opening up new roads for the computational analysis of language, providing researchers with common, generative language structures and syntactic operations, on which language analysis tools were built. A grammar for action would contribute to providing a common framework of the syntax and semantics of action, so that basic tools for action understanding can be built, tools that researchers can use when developing action interpretation systems, without having to start development from scratch.

The input to our system for interpreting manipulation actions is perceptual data, specifically sequences of images and depth maps. Therefore, a crucial part of our system are the vision processes, which obtain atomic symbols from perceptual data. Since hands are the most important components in manipulation actions, a state-of-the-art markerless hand tracking system is used to obtain fine grain skeleton models of both hands. Using this data, the manner in which the human grasps the objects is classified into four primitive categories. On the basis of the grasp classification, the start and end point of action sequences is found. The action then is classified from the hand trajectories, the hand grasp, and the consequence of the object as explained next. To obtain objects, first a contour based attention mechanism locates the object. Next, the manipulated object is monitored using a process that combines stochastic tracking with active segmentation. Then, the segmented objects are recognized. Finally, with the aid of the monitoring process, the effect of the object during action is checked and classified into four types of "consequences" (which are used in the description of the action).

The vision processes produce a set of symbols: the "Subject", "Action" and "Object" triplets, which serve as input to the reasoning module. At the core of our reasoning module is the Manipulation Action Context-Free Grammar (MACFG). This grammar comes with a set of generative rules and a set of parsing algorithms. The parsing algorithms have two main operations: "construction" and "destruction". These algorithms dynamically parse a sequence of tree (or forest) structures made up from the symbols provided by the vision module. The sequence of semantic tree structures could then be used by the cognitive system to perform reasoning and prediction. Fig. 1 shows the flow chart of our proposed cognitive system.

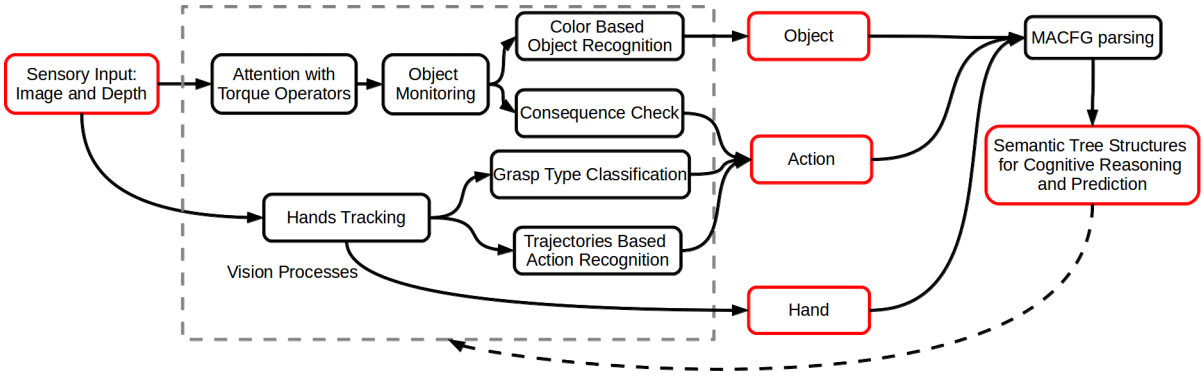


Figure 1: Overview of the proposed cognitive system. Note that there are feedback loops within and between some of the modules (dotted arrows).

2. Related Work

The problem of human activity recognition and understanding has attracted a lot of interest in Computer Vision in recent years. Both visual recognition methods, and the non-visual description methods using Motion capture systems (Guerra-Filho, Fermuller, & Aloimonos, 2005; Li et al., 2010) have been used. A few good surveys of the former can be found in (Moeslund, Hilton, & Krüger, 2006), (Turaga et al., 2008), and (Gavrila, 1999). There is a large range of applications for this work in areas like HCI, biometrics, and video surveillance. The approach of most visual recognition methods is to learn from a large amount of spatio-temporal points what an action looks like ((Laptev, 2005), (Dollár et al., 2005), (Wang & Suter, 2007), (Willems, Tuytelaars, & Van Gool, 2008)). Most of the focus has been on recognizing single human actions like walking, jumping, or running etc. ((Ben-Arie et al., 2002), (Yilmaz & Shah, 2005)). Approaches to more complex actions have employed parametric models such as HMMs (Kale et al., 2004) to learn the transition between image frames (e.g. (Hu et al., 2000; Saisan et al., 2001; Chaudhry et al., 2009) and (Aksoy et al., 2011)).

The problem of understanding manipulation actions is of great interest also in robotics, where usually the focus is on executing these actions. A large amount of work has been devoted to robot learning from demonstration (Argall et al., 2009), for example, the problem of a robot with hands learning to manipulate objects by mapping the real world trajectory observed on people performing the action to the robot body. These kind of approaches have stayed at the level of signal to signal mapping, and lack the ability of generalization. More recently, within the domain of robot control research, (Fainekos, Kress-Gazit, & Pappas, 2005) proposed to use temporal logic for hybrid controller, and later (Dantam & Stilman, 2013) suggested a grammatical formal system to represent and verify robot control policies. (Wörgötter et al., 2012; Aein et al., 2013) proposed to create a library of manipulation actions through semantic object-action relations obtained from visual observation.

(Chomsky, 1993) suggested that a minimalist generative structure, similar to the one in human language, also exists for action understanding. (Pastra & Aloimonos, 2012) introduced a minimalist grammar of action, which defines the set of terminals, features, non-terminals and production rules

for the grammar in the sensorimotor domain. However, this was a purely theoretical description. A first implementation, using as sensory symbols only objects, was implemented in (Summers-Stay et al., 2013). Then (Guha et al., 2013) proposed a minimalist set of atomic symbols to describe the movements in manipulation actions. On the other hand, in the field of natural language understanding, which traces back to 1960s, Schank proposed the Conceptual Dependency (CD) theory (Schank & Tesler, 1969) to represent knowledge acquired from natural language input. In this theory, sentences are represented as a series of diagrams to represent both abstract and real physical actions, which involve agents and objects, and the actions are built from a set of primitive acts, which include atomic symbols like GRASP, MOVE, etc. Also in (Manikonda, Krishnaprasad, & Hendler, 1999), the relationship between languages and motion control is discussed.

In this paper we extend the minimalist action grammar of (Pastra & Aloimonos, 2012) by providing a set of operations based on a set of context-free grammar rules (Table. 1) in order to dynamically parse the observations in a fully automatic manner. Then we provide a set of biological inspired visual processes, which compute from the low level signals the symbols used as input to the grammar in the form of ("Subject", "Action", "Object"). By integrating the perception modules with the reasoning module, we obtain a cognitive system for human manipulation action understanding.

3. The Proposed Cognitive System

In the following sections, we first describe the reasoning module based on the Manipulation Action Context-Free Grammar (3.1) and the parsing algorithms (3.2). Then we discuss the vision methods: the attention mechanism (3.3), the hand tracking and action recognition (3.4), the object monitoring and recognition (3.5), and the action consequence classification (3.6).

3.1 A Context-free Manipulation Action Grammar

In formal language theory, a context-free grammar (CFG) is a formal grammar in which every production rule is of the form $V \rightarrow w$, where V is a single nonterminal symbol, and w is a string of terminals and/or nonterminals (w can be empty). The basic recursive structure of natural languages, the way in which clauses nest inside other clauses, and the way in which lists of adjectives and adverbs are swallowed by nouns and verbs, is described exactly by a context-free grammar..

Similarly, for manipulation actions, every complex activity is built from smaller blocks. Using the notation used in linguistics, a block consists of a a "Subject", "Action" and "Patient" triplet. Here, a "Subject" can be either a hand or an object, and the same holds for the "Patient". Furthermore, a complex activity also has a basic recursive structure, and can be decomposed into simpler actions. For example, the typical manipulation activity "sawing a plank" is described by the top level triplet "handsaw saw plank", and has two lower level triplets (which in time proceed the top level action), namely "hand grasp saw" and "hand grasp plank". Intuitively, the process of observing and interpreting manipulation actions, is syntactically quite similar to natural language understanding. Thus, the following Manipulation Action Context-Free Grammar (MACFG) (Table. 1) is proposed to parse manipulation actions.

AP	$\rightarrow$	$A\ O \mid A\ HP$	(1)
HP	$\rightarrow$	$H\ AP \mid HP\ AP$	(2)
H	$\rightarrow$	h	
A	$\rightarrow$	a	
O	$\rightarrow$	o	(3)

Table 1: Manipulation Action Context-Free Grammar (MACFG).

The nonterminals H, A, and O represent the "hand", the "action" and the "object" (the tools and objects under manipulation), respectively, and the terminals, h, a and o are the observations. AP is the "Action Phrase" and HP is the "Hand Phrase".

The design of this grammar is motivated by the following observations: 1) The main and only driving force in manipulation actions are the hands. Thus, a specialized nonterminal symbol H is used for their representation; 2) An "Action" (A) can be applied to an "Object" (O) directly, or to a "Hand Phrase" (HP), which in turn contains an "Object" (O). This is encoded in Rule (1), which builds up an "Action Phrase" (AP); 3) An "Action Phrase" (AP) can be combined either with the "Hand" (H), or a "Hand Phrase". This is encoded in rule (2), which recursively builds up the "Hand Phrase". The rules above form the syntactic rules of the grammar used in the parsing algorithms.

3.2 Cognitive MACFG Parsing Algorithms

As mentioned in Sec. 1, the goal of this project is not only to propose a grammar for representing manipulation actions, but also to develop a set of operations that can automatically parse (create or dissolve) the semantic tree structures. This is very crucial for practical purposes, since parsing a manipulation action is an on-line process by its nature. The observations are obtained in a temporal sequence. Thus, the parsing algorithm for the grammar should be able to dynamically update the tree structures on the fly. At any point in time the current leaves of the semantic forest structures represent the actions and objects involved up to now. When a new triplet of ("Subject", "Action", "Patient") arrives, the parser updates the tree using one of two possible operations described below.

Theoretically the non-regular context-free language define in Table. 1 can be recognized by a non-deterministic pushdown automaton. However, different from language input, the perception input is naturally a sequence of key observations which has orders. Thus, instead of simply building a non-deterministic pushdown automaton, it requires a special set of parsing operations.

Our parsing algorithm differentiates between two kinds of "Actions" A: constructive and destructive actions. Constructive actions are the movements that start with the hand (or a tool) coming in contact with an object and usually result in a certain physical change on the object (a consequence), e.g. "Grasp", "Cut", or "Saw". Destructive actions are the movements at the end of physical change inducing actions, when the hand (or tool) separates from the object; some examples are "Ungrasp" or "FinishedCut". A constructive action may or may not have a corresponding destructive action, but every destructive action must have a corresponding constructive action. Otherwise the parsing algorithm should detect an error. In order to facilitate the action recognition, a look-up table that stores the (constructive - destructive) action pairs serves as knowledge. This knowledge can be learned and further expanded.

The algorithm builds a tree group (forest) T_s, that is updated in time as new observation are received. Once an observation triplet "Subject", "Action" and "Patient" comes in, the algorithm first checks whether the "Action" is constructive or destructive, and then follows one of two pathways (see Algorithm. 1). If the "Action" is constructive, a CONSTRUCTION() routine is called (see details in Algorithm. 2). Otherwise a DESTRUCTION() routine is called (see details in Algorithm. 3). The process continues till the last observation. Two graphical illustrations in Fig 2 demonstrate how the CONSTRUCTION() and the DESTRUCTION() routines work. In computer science terminology, the PARSE() operation amounts to a chart-based parser (Younger, 1967), which takes in the three nonterminals and performs bottom-up parsing following the context-free rules from Table 1.

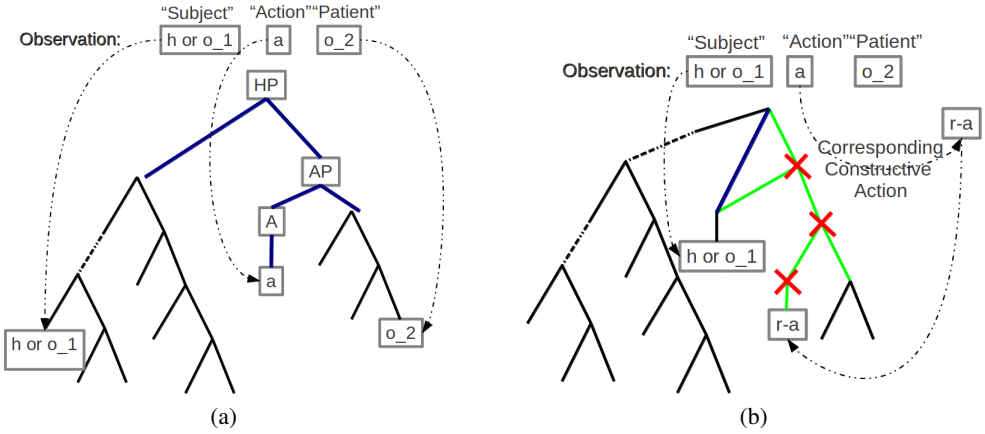


Figure 2: The (a) CONSTRUCTION() and (b) DESTRUCTION() operations. Blue lines are newly added connections, red cross means node deletion and green lines are connections to be deleted.

In Fig. 3, a typical manipulation action example is given. The parsing algorithm described here takes as input a sequence of key observations: "LeftHand Grasp Knife; RightHand Grasp Eggplant; Knife Cut Eggplant; Knife FinishedCut Eggplant; RightHand Ungrasp Eggplant; LeftHand Ungrasp Knife". Then a sequence of six tree structures is automatically bottom-up parsed or dissolved along the time-line. More examples can be found in Sec. 4, and sample implementation of the parsing algorithm is provided at (`http://www.umiacs.umd.edu/~yzyang/MACFG`). In the figure for better illustration we use a DUMMY root node and we number the nonterminal nodes.

3.3 Attention Mechanism Using the Torque Operator

It is essential for a cognitive system to have an effective attention mechanism, because the amount of information in real world images is enormous. Visual attention, the process of driving an agent's attention to a certain area, is based on both bottom-up processes defined on low level visual features, and top-down processes influenced by the agent's previous experience (Tsotsos, 1990). Recently, (Nishigaki, Fermuller, & DeMenthon, 2012) proposed a vision tool, called the image torque, which using bottom-up processes, captures the concept of closed contours. Basically, the torque operator takes as input edges and computes over regions of different size a measure of how well the edges are aligned to form a closed, convex contour.

Algorithm 1 Dynamic Manipulation Action Tree Parsing

Initialize an empty tree group (forest) T_s
while New observation (subject s, action a, patient p) **do**
  if a is a constructive action **then**
    CONSTRUCTION(T_s, s, a, p)
  end if
  if a is a destructive action **then**
    DESTRUCTION(T_s, s, a, p)
  end if
end while

Algorithm 2 CONSTRUCTION(T_s, s, a, p)

Previous tree group (forest) T_s and new observation (subject s, action a and patient p)
if s is Hand h, and p is an object o **then**
  Find the highest subtrees T_h and T_o from T_s containing h and o. If h or o is not in the current forest, create new subtrees T_h and T_o respectively.
  PARSE(T_h,a,T_o), attach it to update T_s.
end if
if s is an object o_1 and p is another object o_2 **then**
  Find the highest subtree T_o^1 and T_o^2 from T_s containing o_1 and o_2 respectively. If either o_1 or o_2 is not in the current forest, create new subtree T_o^1 or T_o^2. If both o_1 and o_2 are not in the current forest, raise an error.
  PARSE(T_o^1,a,T_o^2), attach it to update T_s.
end if

Algorithm 3 DESTRUCTION(T_s, s, a, p)

Previous tree group (forest) T_s and new observation (subject s, action a and patient p)
Find corresponding constructive action of a from the look-up table, denote it as a'
if There exists a lowest subtree T_a' containing both s and a' **then**
  Remove every node on the path starting from root of T_a' to a'.
  if T_a' has parent node **then**
    Connect the highest subtree that contains s with T_a''s parent node.
  end if
  Leave all the remaining subtrees as individual trees.
end if
Set the rest of T_s as new T_s.

In this work, we pre-process the testing sequences by applying the torque operators to obtain possible initial fixation points for the object monitoring process. An example showing the application of the torque operator is shown in Fig. 4.

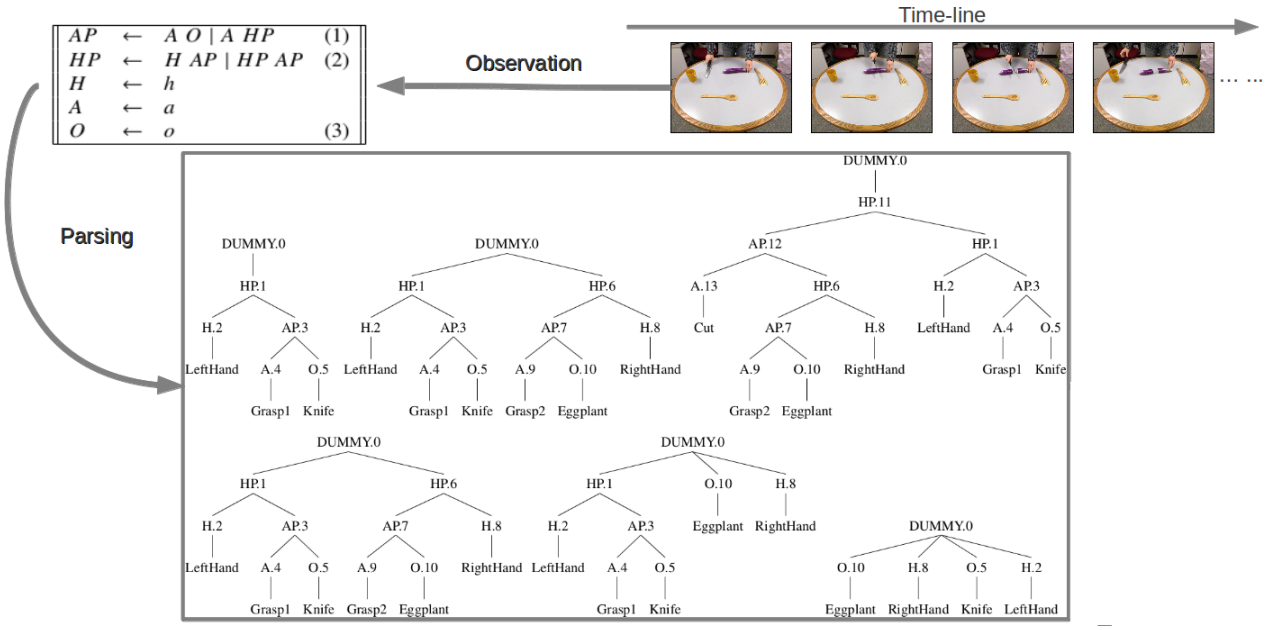
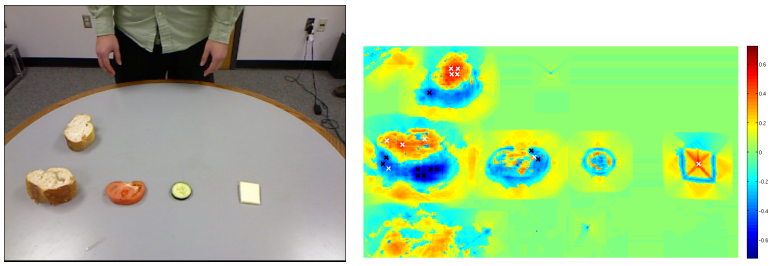


Figure 3: The framework for the MACFG and its associated parsing algorithms.



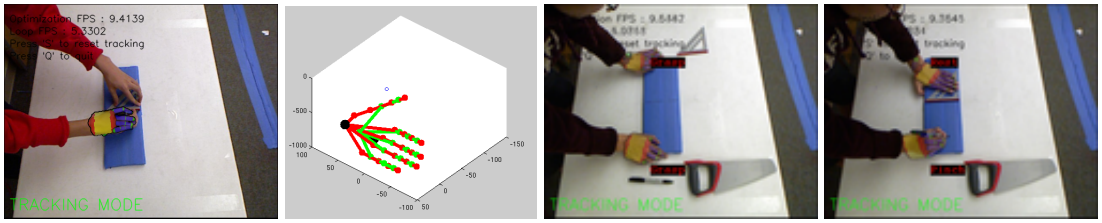
(a) (b)

Figure 4: Torque operator response (b) on a sample input frame (a). White and black crosses are the potential fixation points.

The system also employs some form of top-down attention mechanism; it uses the hand location to guide the attention. In our approach we integrate the bottom-up torque operator output with hand tracking. Potential objects under manipulation are found, when one of the hand regions intersects a region with high torque responses, and then the object monitoring system is called to monitor it.

3.4 Hand Tracking, Grasp Type Classification and Trajectory based Action Recognition

With the recent development of a vision based, markerless, fully articulated model based human hand tracking system (`http://cvrlcode.ics.forth.gr/handtracking/`) (Oikonomidis, Kyriazis, & Argyros, 2011), we are able to track a 26 degree of freedom hand model in real-time. While for a simple classification of movements into a small number of actions, the location of the



(a) (b) (c) (d)

Figure 5: (a): one example of fully articulated hand model tracking, (b): a 3-D illustration of the tracked model. Red dots are fingertips and joints, green dots are corresponding bones, black dot is the wist and the palm center. (c)(d): examples of grasp type recognition for both hands.

hands and objects would be sufficient, with the full hand model a finer granularity of description can be achieved by classifying the tracked hand-model into different grasp types.

We collected training data from different actions, which then was processed as follows: A set of bio-inspired features (following the hand anatomy (Tubiana, 1998)) were extracted. Intuitively, the arches formed by the fingers are crucial to differentiate different grasp types. The fixed and mobile parts of the hand adapt to various everyday tasks by forming bony arches: longitudinal arches (the rays formed by the finger bones and their associated metacarpal bones), and oblique arches (between the thumb and four fingers). For more information, refer to `http://en.wikipedia.org/wiki/Hand`.

In each image frame, we computed the oblique and longitudinal arches (see Fig. 5b) to obtain a 8 parameter feature vector. We further reduced the dimensionality of the feature space by PCA and then applied k-means clustering to discover (from the collected data) the underlying four general types of grasp, which are 1)REST, 2)FIRM GRASP, 3)DELICATE GRASP (PINCH) and 4)EXTENSION GRASP. To classify a given test sequence, the data was processed as described above and then the grasp type was computed using a Naive-Bayes classification (See Fig. 5c 5d).

The grasp classification is used to segment the image sequence in time, and as part of the action description. In addition, we also use the trajectory of the mass center of the hands to classify the actions. The hand tracking software provides the hand trajectories (of the given action sequence between the onset of grasp and release of the object), from which we computed global features of the trajectory (frequency, velocity) which were then used for classification. For more details, refer to (Teo et al., 2012).

3.5 Object Monitoring and Recognition

Here we apply a new method combining segmentation and tracking to obtain the information necessary to monitor the objects under influence (Yang, Fermuller, & Aloimonos, 2013). This method combines stochastic tracking (Han et al., 2009) with a fixation based active segmentation (Mishra, Fermuller, & Aloimonos, 2009). The tracking module provides a number of tracked points. The locations of these points are used to define an area of interest and a fixation point for the segmentation, and the color in their immediate surroundings are used in the data term of the segmentation

module. The segmentation module segments the object, and based on the segmentation, updates the appearance model for the tracker. Using this method we are able to track objects as they deform and change topology (two objects can merge, or an object can be divided into parts.) Fig 6 illustrates the method over time.

The proposed method is a dynamically closed-loop process, where the active segmentation provides the target model for next tracking step, while the stochastic tracking provides the attention field for the active segmentation.

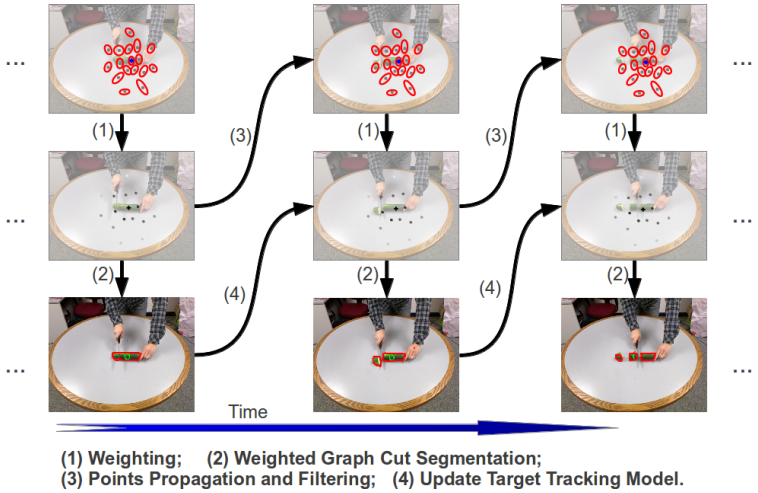


Figure 6: Flow chart of the active segmentation and tracking method for object monitoring.

For object recognition, we simply used color. Since the objects in our experiments have distinct color profiles, the color distribution model that is used as feature was sufficient for recognizing the segmented objects. We manually labelled several examples of each object class as training data, and used a nearest neighbour classification. Sample results are shown in Fig. 8.

3.6 Manipulation Action Consequence Check

Taking an object-centric point of view, manipulation actions can be classified into six categories according to how the object is transformed during the manipulation, or in other words what consequence the action has on the object. These categories are: DIVIDE, ASSEMBLE, CREATE, CONSUME, TRANSFER, and DEFORM.

To describe these action categories we need a formalism. We use the visual semantic graph (VSG) inspired from the work of (Aksoy et al., 2011). This formalism takes as input computed object segments, their spatial relationship, and temporal relationship over consecutive frames. Every frame is described by a Visual Semantic Graph (VSG), which is represented by an undirected graph $G(V,E,P)$. The vertex set $|V|$ represents the set of semantically meaningful segments, the edge set $|E|$ represents the spatial relations between any of the two segments. Two segments are connected when they share parts of their borders, or when one of the segments is contained in the other. If two nodes $v_1, v_2 \in V$ are connected, $E(v_1,v_2) = 1$, otherwise, $E(v_1,v_2) = 0$. In addition, every

node $v \in V$ is associated with a set of properties $P(v)$, that describes the attributes of the segment. This set of properties provides additional information to discriminate the different categories, and in principle many properties are possible. Here we use location, shape, and color.

We need to compute the changes in the object over time. In our formulation this is expressed as the change in the VSG. At any time instance t, we consider two consecutive VSGs, the VSG at time $t-1$, denoted as $G_a(V_a, E_a, P_a)$ and the VSG at time t, denoted as $G_z(V_z, E_z, P_z)$. We then define the following four consequences, where $\leftarrow$ is used to denote the temporal correspondence between two vertices, $\nrightarrow$ is used to denote no correspondence:

- DIVIDE: $\{\exists v_1 \in V_a; v_2, v_3 \in V_z | v_1 \leftarrow v_2, v_1 \leftarrow v_3)\}$ *or* $\{\exists v_1, v_2 \in V_a; v_3, v_4 \in V_z | E_a(v_1, v_2) = 1, E_z(v_3, v_4) = 0, v_1 \leftarrow v_3, v_2 \leftarrow v_4\}$ **Condition (1)**
- ASSEMBLE: $\{\exists v_1, v_2 \in V_a; v_3 \in V_z | v_1 \leftarrow v_3, v_2 \leftarrow v_3\}$ *or* $\{\exists v_1, v_2 \in V_a; v_3, v_4 \in V_z | E_a(v_1, v_2) = 0, E_z(v_3, v_4) = 1, v_1 \leftarrow v_3, v_2 \leftarrow v_4\}$ **Condition (2)**
- CREATE:$\{\forall v \in V_a; \exists v_1 \in V_z | v \nrightarrow v_1\}$ **Condition (3)**
- CONSUME:$\{\forall v \in V_z; \exists v_1 \in V_a | v \nrightarrow v_1\}$ **Condition(4)**

While the above actions can be defined purely on the basis of topological changes, there are no such changes for TRANSFER and DEFORM. Therefore, we have to define them through changes in property. In the following definitions, P^L represents properties of location, and P^S represents properties of appearance (shape, color, etc.).

- TRANSFER:$\{\exists v_1 \in V_a; v_2 \in V_z | P_a^L(v_1) \neq P_z^L(v_2)\}$ **Condition (5)**
- DEFORM: $\{\exists v_1 \in V_a; v_2 \in V_z | P_a^S(v_1) \neq P_z^S(v_2)\}$ **Condition (6)**

A graphical illustration for **Condition (1-6)** is shown in Fig. 7. The active object monitoring process introduced in Sec. 3.5 to 1) is used to find correspondences ($\leftarrow$) between V_a and V_z; 2) monitor location property P^L and appearance property P^S in the VSG.

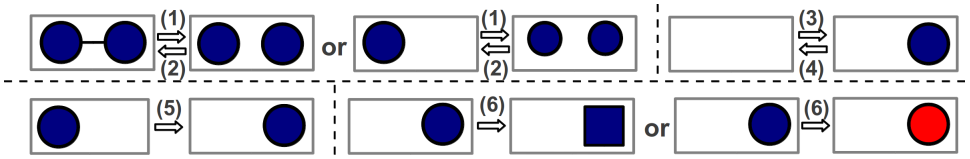


Figure 7: Graphical illustration of the changes for **Condition (1-6)**.

4. Experiments

To validate our system, RGBD sequences of a person performing different manipulation actions were collected. The actions were: "Making a sandwich", "Cutting an eggplant", "Cutting Bread" and "Sawing a plank into two pieces". The list of "Subjects", "Objects", and "Actions" involved in our experiments is given in Table. 2.

Given the sequence of triplets computed by the vision modules (Sec. 3.3 3.4 3.5 3.6), a sequence of trees (or forests) are created or dissolved dynamically. To be specific, the grasp type classification module provides a "Grasp" signal when the hand status change from "REST" to one of the three types, and an "Ungrasp" signal when it changes to "REST". At the same time, the object monitoring and the segmentation based object recognition module provide the "Object" symbol when either of

Hand	*Object*	*Constructive Action*	*Destructive Action*
LeftHand	*Bread*, *Cheese*, *Tomato*, *Eggplant*,	*Grasp*, *Assemble*,	*Ungrasp*, *FinishedAssemble*,
RightHand	*Plank*, *Saw*, *Cucumber*,	*Cut*, *Saw*,	*FinishedCut*, *FinishedSaw*,

Table 2: "Hands", "Actions" and "Objects" involved in the experiments.

the hands touches an object. Also the hand tracking module provides trajectory profiles that enable the trajectory based action recognition module to produce "Action" symbols like "Cut" and "Saw". The action "Assemble" does not have a distinctive trajectory profile, so we simply generate it when the "Cheese" merges with the "Bread" based on the object monitoring process. At the end of each recognized action, a corresponding destructive symbol, as defined in Table. 2, is produced, and the consequence checking module is called to confirm the action consequence. Fig. 8 shows intermediate and final results from a sequence of a person making a sandwich. More results on a variety of manipulation actions can be found at `http://www.umiacs.umd.edu/~yzyang/MACFG`.

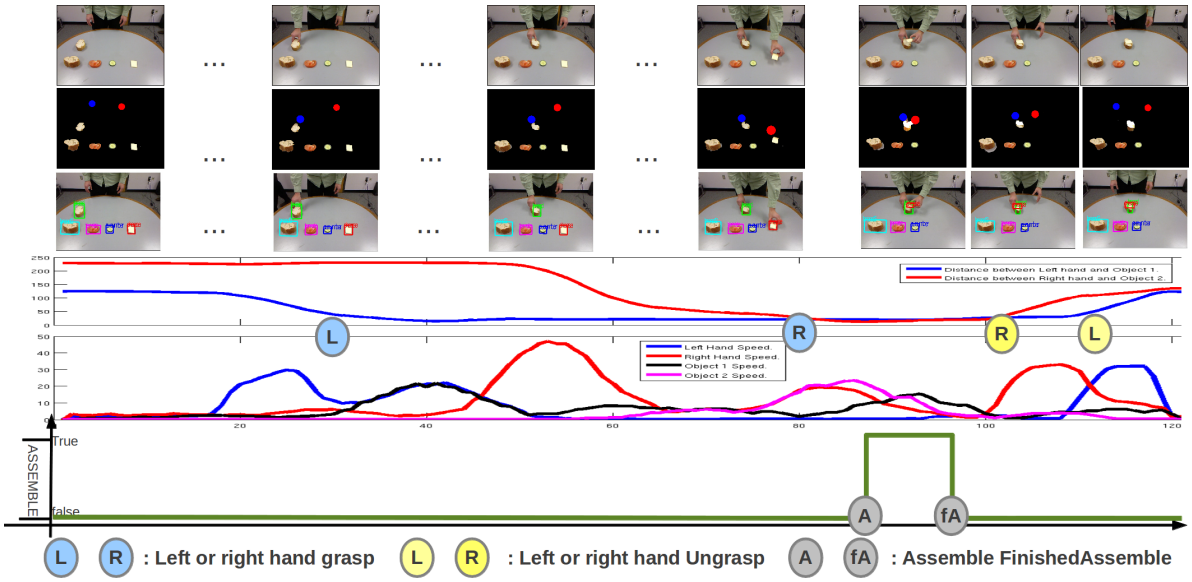
Our experiments showed that the proposed cognitive system was able to successfully update the semantic structures over time. Also it was found flexible enough to handle complex situations like 1) hesitation: the human grasps a tool and finds out that it is not the desired one and ungrasps it (for example, see Fig. 9); 2) multiple people manipulating on the same workspace; 3) deep recursion: a hand grasps object 1, which serves as tool to perform an action on object 2 and so on.

5. Conclusion and Future Work

A cognitive system for human manipulation action understanding was presented. The system integrates vision modules that ground semantic meaningful events in perceptual input with a reasoning module based on a context-free grammar and associated parsing algorithms, which dynamically build the sequence of structural representations (trees or forests). Experimental results showed that the proposed cognitive system can extract the key events from the raw input, and can interpret the observations by generating a sequence of trees or forests.

In future work we will further generalize our framework. First, since the grammar is context-free, a direct extension is to make it probabilistic. Second, since the grammar does not assume constraints like the number of operators, it can be further adapted to process scenarios with multiple agents doing complicated manipulation actions, once the perception tools have been developed.

Moreover, we also plan to investigate operations which enable the cognitive system to reason during observation. After the system observes a significant number of manipulation actions, it can build a database of all the sequences of trees. By querying this database, the cognitive system can be expected to predict things like which object will be manipulated next or which action will follow. Also, the action trees could be learned not only from observation but also from language resources (dictionaries, recipes, manuals, etc.). This possibility bridges cognitive system research to computational linguistics and constitutes an interesting research avenue.



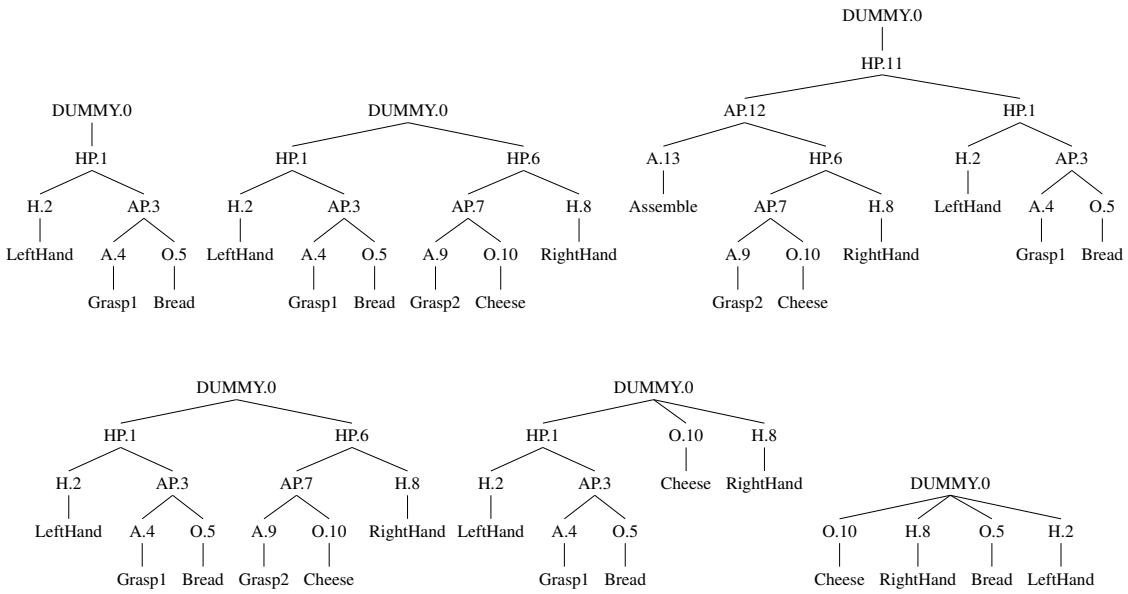


Figure 8: Experimental result: Row two shows the hand tracking and object monitoring. Row three shows the object recognition result, where each segmentation is labelled with an object name and a bounding box in different color. Row four and five depict the hand speed profile and the Euclidean distances between hands and objects. Row six shows the consequence detection. Row seven and eight are the tree structures generated.

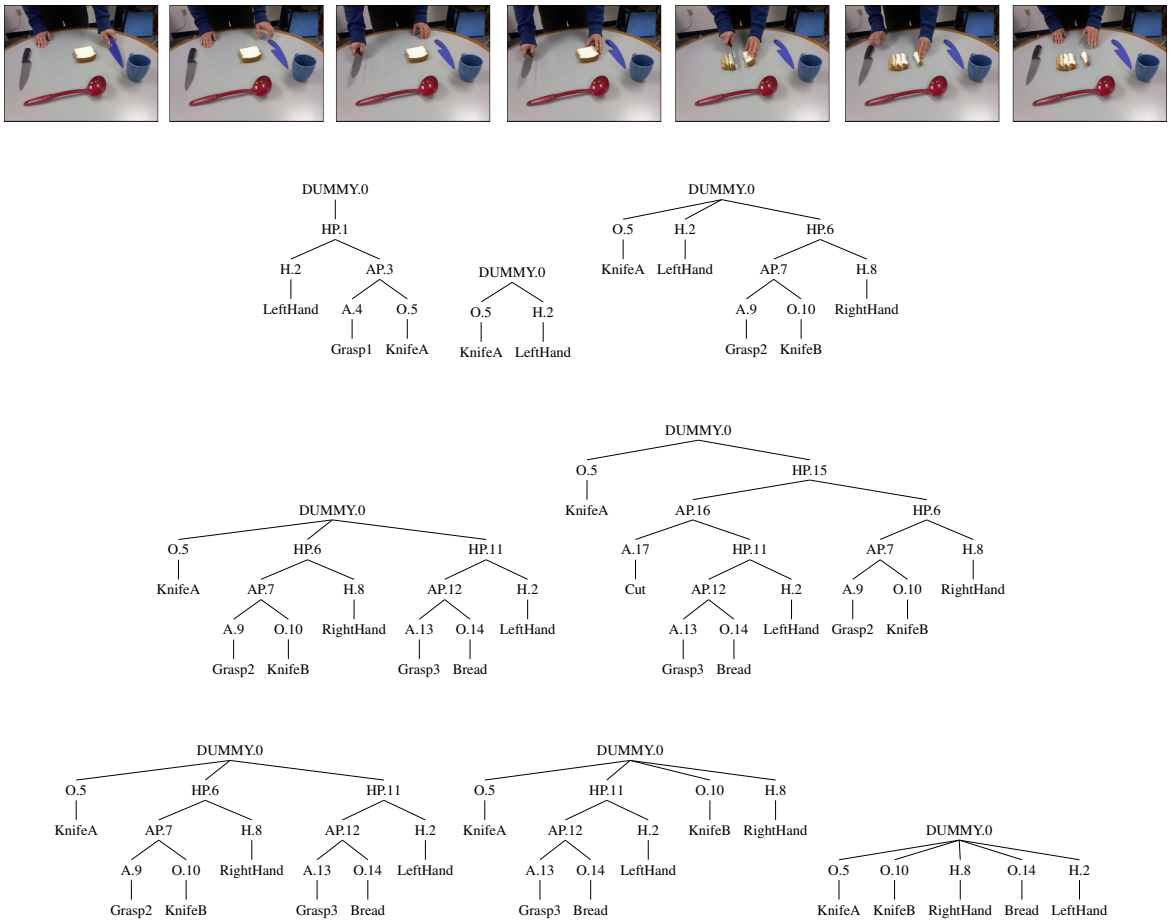


Figure 9: Example of the grammar that deals with hesitation. Key frames of the input image data and the semantic tree structures are shown here.

References

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

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

Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, *57*, 469–483.

Ben-Arie, J., Wang, Z., Pandit, P., & Rajaram, S. (2002). Human activity recognition using multidimensional indexing. *PAMI, IEEE Transactions on*, *24*, 1091–1104.

Chaudhry, R., Ravichandran, A., Hager, G., & Vidal, R. (2009). Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on* (pp. 1932–1939).

Chomsky, N. (1957). *Syntactic structures*. Mouton de Gruyter.

Chomsky, N. (1993). *Lectures on government and binding: The pisa lectures*, Vol. 9. Walter de Gruyter.

Dantam, N., & Stilman, M. (2013). The motion grammar: Analysis of a linguistic method for robot control. *Transactions on Robotics (T-RO)*.

Dollár, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on* (pp. 65–72).

Fainekos, G. E., Kress-Gazit, H., & Pappas, G. J. (2005). Hybrid controllers for path planning: A temporal logic approach. *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on* (pp. 4885–4890).

Gavrila, D. M. (1999). The visual analysis of human movement: A survey. *Computer vision and image understanding*, *73*, 82–98.

Guerra-Filho, G., Fermuller, C., & Aloimonos, Y. (2005). Discovering a language for human activity. *Proceedings of the AAAI 2005 fall symposium on anticipatory cognitive embodied systems, Washington, DC*.

Guha, A., Yang, Y., Fermüller, C., & Aloimonos, Y. (2013). Minimalist plans for interpreting manipulation actions. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Han, B., Zhu, Y., Comaniciu, D., & Davis, L. (2009). Visual tracking by continuous density propagation in sequential bayesian filtering framework. *PAMI, IEEE Transactions on*, *31*, 919–930.

Hu, C., Yu, Q., Li, Y., & Ma, S. (2000). Extraction of parametric human model for posture recognition using genetic algorithm. *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on* (pp. 518–523).

Kale, A., Sundaresan, A., Rajagopalan, A., Cuntoor, N., Roy-Chowdhury, A., Kruger, V., & Chellappa, R. (2004). Identification of humans using gait. *Image Processing, IEEE Transactions on*, *13*, 1163–1173.

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, *64*, 107–123.

Li, Y., Fermuller, C., Aloimonos, Y., & Ji, H. (2010). Learning shift-invariant sparse representation of actions. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (pp. 2630–2637).

Manikonda, V., Krishnaprasad, P. S., & Hendler, J. (1999). *Languages, behaviors, hybrid architectures, and motion control*. Springer.

Mishra, A., Fermuller, C., & Aloimonos, Y. (2009). Active segmentation for robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Moeslund, T., Hilton, A., & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, *104*, 90–126.

Nishigaki, M., Fermuller, C., & DeMenthon, D. (2012). The image torque operator: A new tool for mid-level vision. *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on* (pp. 502–509).

Oikonomidis, I., Kyriazis, N., & Argyros, A. (2011). Efficient model-based 3d tracking of hand articulations using kinect. *British Machine Vision Conference*.

Pastra, K., & Aloimonos, Y. (2012). The minimalist grammar of action. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *367*, 103–117.

Saisan, P., Doretto, G., Wu, Y., & Soatto, S. (2001). Dynamic texture recognition. *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on* (pp. II–58).

Schank, R. C., & Tesler, L. (1969). A conceptual dependency parser for natural language. *Proceedings of the 1969 conference on Computational linguistics* (pp. 1–3).

Summers-Stay, D., Teo, C., Yang, Y., Fermuller, C., & Aloimonos, Y. (2013). Using a minimal action grammar for activity understanding in the real world. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Teo, C., Yang, Y., Daume, H., Fermuller, C., & Aloimonos, Y. (2012). Towards a watson that sees: Language-guided action recognition for robots. *IEEE International conference on Robotics and Automation* (pp. 374–381).

Tsotsos, J. (1990). Analyzing vision at the complexity level. *Behavioral and brain sciences*, *13*, 423–469.

Tubiana, R. (1998). *Examination of the hand & the wrist*.

Turaga, P., Chellappa, R., Subrahmanian, V., & Udrea, O. (2008). Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, *18*, 1473–1488.

Wang, L., & Suter, D. (2007). Learning and matching of dynamic shape manifolds for human action recognition. *Image Processing, IEEE Transactions on*, *16*, 1646–1661.

Willems, G., Tuytelaars, T., & Van Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. *IEEE European Conference on Computer Vision*, 650–663.

Wörgötter, F., Aksoy, E. E., Kruger, N., Piater, J., Ude, A., & Tamosiunaite, M. (2012). A simple ontology of manipulation actions based on hand-object relations. *Autonomous Mental Development, IEEE Transactions on*.

Yang, Y., Fermuller, C., & Aloimonos, Y. (2013). Detection of manipulation action consequences (mac). *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*.

Yilmaz, A., & Shah, M. (2005). Actions sketch: A novel action representation. *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on* (pp. 984–989).

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n^3. *Information and control*, *10*, 189–208.