Manipulation Action Tree Bank: A Knowledge Resource for Humanoids

Yezhou Yang[1], Anupam Guha[1], Cornelia Fermüller[2], and Yiannis Aloimonos[1]

Abstract— **Our premise is that actions of manipulation are represented at multiple levels of abstraction. At the high level a grammatical structure represents symbolic information (objects, actions, tools, body parts) and their interaction in a temporal sequence, and at lower levels the symbolic quantities are grounded in perception. In this paper we create symbolic high-level representations in the form of manipulation action tree banks, which are parsed from annotated action corpora. A context free grammar provides the grammatical description for the creation of the semantic trees. Experiments conducted on the tree banks show that they allow to 1) generate so-called visual semantic graphs (VSGs), 2) compare the semantic distance between steps of activities and 3) discover the underlying semantic space of an activity. We believe that tree banks are an effective and practical way to organize semantic structures of manipulation actions for humanoids applications. They could be used as basis for 1) automatic manipulation action understanding and execution and 2) reasoning and prediction during both observation and execution. The knowledge resource follows the widely used Penn Tree Bank format.**

I. Introduction

Autonomous humanoid robots need to have the capability to understand, learn and execute actions of manipulation. These actions involve objects and tools, are composed of primitive sub-actions, and are performed using the robots' effectors. Recent studies on humans have shown that grammatical structures underlie our representation of manipulation actions, which are used both to understand and to execute these actions. Understanding manipulation actions is like understanding language, while executing them is like generating language.

If a household robot is innately wired with the capability of understanding and generating language, it should be able to apply a similar mechanism to understand and execute manipulation actions. It should be able to learn how to do an action from either observing a human doing it (learn from observation) or from parsing human instructions (learn from language). However, given that there is effectively an infinite number of ways to do a certain manipulation action, such as for example, making a peanut butter and jelly sandwich [1], the robot should be able to store and organize all the semantic structures effectively, rather than simply an enumeration of all possible ways. Moreover, assuming that the semantic structure has been stored properly in the robot memory, further problems of doing the prediction and reasoning over them effectively also need to be solved.

[1]Y. Yang, A. Guha, and Y. Aloimonos, are from the Department of Computer Science, and [2]C. Fermüller is from UMIACS, University of Maryland, College Park, MD 20742, USA {yzyang, aguha, yiannis} at cs.umd.edu and fer at umiacs.umd.edu

In the field of computational linguistics, parsed text corpora with annotations presented as compositional tree structures, also called tree banks [2] have proven to be the most effective way to organize syntactic and even semantic structures of natural languages. The typical way of building a tree bank involves: 1) first humans annotate each sentence in the corpus extensively, 2) then parse them into grammar trees (or other more sophisticated structures). These annotated tree banks can serve as learning bases for automatic sentence part of speech (POS) tagging and bottom up parsing models. Later these models can be used to automatically tag and parse more natural language resources in order to bootstrap the whole tree bank. Prediction and reasoning based on tree banks has been extensively studied in the field of computational linguistics.

Similarly, in this paper, we propose that tree banks are an effective and practical way to organize semantic structures of manipulation actions for robotics. They could be used as basis for 1) automatic manipulation action understanding and execution and 2) doing reasoning and prediction during both observation and execution, as illustrated in Fig. 1.

The rest of the paper is organized as follows: In Sec. II a brief survey of related work is provided. In Sec. III we will briefly introduce the manipulation action context-free grammar that is used to parse semantic tree structures in this work. In Sec. IV the technical details for building the manipulation action tree banks are presented. In Sec. V two sets of publicly available manipulation action datasets are used to build different levels of tree banks, three experiments are conducted, and one example of using the knowledge resource is given. Sec. VI describes conclusion and future works. The tree banks can be downloaded from `http://www.umiacs.umd.edu/~yzyang/MA_Tree_Bank/`.

II. Related Work

Robotic manipulation has attracted a great amount of interest due to its direct applications in intelligent manufacturing. With the recent development of advanced robotic manipulators, work on robust and accurate manipulation techniques has followed quickly. For example, [3] developed a method for the PR2 to fold towels, which is based on visual processes of grasp point detection and multiple-view geometry. [4] and [5] developed for their humanoid robot, ARMAR-III, manipulation and perception capabilities based on imitation learning for human environments. [6] proposed learning object affordances models in multi-object manipulation tasks. [7] investigated robots searching for objects using reasoning about both perception and manipulation. A good survey on humanoid dual arm manipulation can be found in [8]. These

works achieved promising results on robotic manipulation, but they focused on specific actions, and do not allow for generalization. Here we propose that manipulation action tree banks, by capturing the syntactic structure of manipulation actions, provide a general solution to organize the underlying semantics behind specific actions and allow the robot to do more sophisticated tasks like reasoning and prediction.

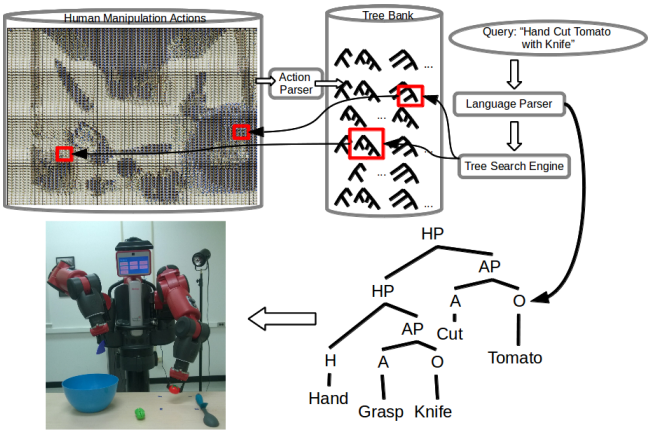


Fig. 1: A robotic system using a manipulation tree bank.

Human activity recognition and understanding has been studied heavily in Computer Vision recently and there is a large range of applications for this work in areas like human-computer interactions, biometrics, and video surveillance. Both visual recognition methods, and the non-visual description methods using motion capture systems have been used. Surveys of the former can be found in [9] and [10].

There has been some work on the way instructions are stored and analyzed for manipulation actions, generally as sequences. Work done by [11] among others investigates how these sequence datasets can be processed in order to reason about manipulation actions. Sequence alignment borrows techniques from informatics. Our paper deals with grammar trees, a more detailed representation of manipulation actions. Chomsky in [12] suggested that a minimalist generative grammar, similar to the one of human language, also exists for action understanding and execution. The works closest related to this paper are [13], [14], [15], [16]. [13] first discussed a Chomskyan grammar for understanding complex actions as a theoretical concept, [14] provided an implementation of such a grammar using as perceptual input only objects, and Guha et al. [15] suggested a set of atomic symbols for describing movement.[16] proposed a set of context-free grammar rules for manipulation action understanding, as well as a set of operations to parse the subject, action, object triplets into grammatical tree structures. Since the tree banks are created using the methods from [16], we will briefly introduce the grammar and parsing rules in Sec. III.

In the field of computational linguistics, there have been attempts to train robots using natural language [17] by identifying robot primitives from natural language utterances. The most explored area has been robot navigation using natural language instructions in works like [18], [19] where semi-structured environment, grounded knowledge etc. are used by the robot to infer actions from natural language commands. Attempts to use natural language tree like structures is less common but present in works like [20] where parsing of natural language instructions plays an important part in generating tree like plans for the robot. The logical extension of these works is to make tree banks of the actions themselves and use them to reason about actions in a language like manner. Also, in [21] a probabilistic first-order knowledge base was built as action-specific knowledge for robots acting in household environments. In this paper we further focus on manipulation actions, and propose to build tree banks for manipulation actions to serve as a hierarchical knowledge base for autonomous humanoids to effectively perform reasoning and prediction over the learned semantic structures. We did experiments on two sets of datasets at different levels of granularity.

III. A Grammatical Representation of Manipulation Actions

In [16] a context-free grammar was proposed for robots to understand human manipulation actions. Every complex activity is built from smaller blocks, where a block is 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". The following Manipulation Action Context-Free Grammar (MACFG) (Table. I) was proposed to parse manipulation actions, and in this work we follow the same grammatical rules.

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 I: **M**anipulation **A**ction **C**ontext-**F**ree **G**rammar.

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

The design of this grammar is motivated by these observations: First, the main and only driving force in manipulation actions are the hands. Thus, a specialized nonterminal symbol H is used for their representation. Second, 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). Finally, 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.

To automatically parse the key semantic triplets (subject, action, patient) into tree structures, two operations, CONSTRUCTION() and DESTRUCTION() were proposed also. For more details, please refer to [16].

IV. Manipulation Action Tree Bank

The concept of a "tree bank" is widely used in the field of computational linguistics, such as the Penn Treebank [2]. Different from natural language tree banks, where the basic unit is a parsed grammar tree for each tagged sentence, each tree for manipulation actions, represents the current status of the action. Since every primitive action in a complex manipulation activity comes in a temporal order, for each sequence, a list of trees are created. For example, Fig. 2 shows a typical set of trees that represent the action "Cutting a Tomato and Put Tomato into Bowl".

Thus for each manipulation action S_n, with $n \in (1...N)$ in the dataset D_m, a list of trees T_n is created. $T_n = \{t_1^n, t_2^n ... t_k^n\}$ where k is the number of key semantic triplets involved in the activity. In Sec. V, two tree banks at different levels of semantic granularity are created from two publicly available and annotated manipulation action datasets.

V. Datasets and Experiments

A. Datasets

Cooking actions are a typical group of manipulation actions, and we use such actions here as testing-bed for both action understanding and execution. To test manipulation action tree banks, we use two state-of-the-art fully annotated cooking datasets, namely the 50 Salad Dataset [22] and the TACoS Cooking Dataset [23]. Though they are both cooking datasets, each of them has its own focused set of actions, which makes them a great complement to each other. A brief introduction for both datasets is given below. Table. II lists the technical details for each of them.

1) 50 Salad Dataset: The 50 Salad dataset captures 25 people preparing 2 mixed salads each and contains over four hours of annotated accelerometer and RGB-D video data. Including detailed annotations, multiple sensor types, and two sequences per participant, the 50 Salads dataset can be used for research in areas such as activity recognition, activity spotting, sequence analysis, progress tracking, sensor fusion, transfer learning, and user-adaptation.

The 50 Salad dataset has its specific focus on salad making, capturing the different variations of one typical manipulation action. Before parsing the (subject, action, patient) annotations into trees, we first automatically extend the original annotations with context sub-actions like "hand grasp knife" and "hand ungrasp knife". Since the annotations provided from the 50 salad dataset comes with "pre" and "post" tags, it can be directly used to indicate constructive and destructive actions required by the parsing algorithm. Examples of grammatical trees built from sequences of this dataset are illustrated in Fig. 2.

2) TACoS Cooking Dataset: The Saarbrücken Corpus of Textually Annotated Cooking Scenes (short: TACoS) contains a set of video descriptions (in natural language) and timestamp-based alignment with the videos. The videos along with a low-level activity annotation is available as part of the corpus. It contains 127 kitchen sequences containing 41 different activities ranging from "cut a cucumber" to "cook carrot soup". 60 different action labels are used for annotation (e.g. PEEL, STIR, TRASH). Also a location tag is accompanied with each (subject, action, patient) triplet to further modify actions. A straightforward extension of the grammar (Table. I) by adding in new rules (Table. III) with a nonterminal "ADV" (adverb), is used to accommodate the extra information (From location A To location B, or At location A). Also, since there is no annotation of destructive actions, we adapt the parsing schema without the DESTRUCTIVE() routine.

A	$\rightarrow$	$A\ ADV$	(1)
ADV	$\rightarrow$	$From/At\ To$	(2)

TABLE III: Extra MACFG rules for location information.

Final grammatical trees built from the first sequence in the TACoS dataset are illustrated in Fig. 3. Note that all the non-terminals are emitted to save space.

B. From Manipulation Action Trees to VSGs

Manipulation action trees can be used to generate Visual Semantic Graphs (VSGs). A visual semantic graph (VSG), proposed in the work of [24], provides a plausible formalism for semantic activity representation. 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, $E(v_1, v_2) = 0$.

The VSG can be generated from the manipulation action tree sequences by following two rules: 1) replace constructive actions and destructive actions with connected lines and disconnected lines; 2) keep the connected lines if the action is a MERGE action, such as PLACE or PUT. Fig. 2 shows corresponding manipulation action trees with their VSGs at each key frame.

C. Tree Edit Distance Analysis

Manipulation action trees can be used to compare the semantic distance between different steps of the activity. In order to compare the trees, we use a measure of tree edit distance. The edit distance between two trees is the minimal-cost sequence of edit operations on labeled nodes that transforms one tree into the other. The following operations are possible, each with its own cost:

Dataset	*#Seq*	*#Events*	*Ave. #Events/Seq*	*#Activities*	*#Actions*	*#People*	*Ave. Length*
50 *Salad*	54	3820	71	1 (*salad making*)	17	25	10703 *frames*
TACoS	127	5530	44	41	60	22	4.5 *min*

TABLE II: Technical details of the 50 Salad and TACoS datasets.

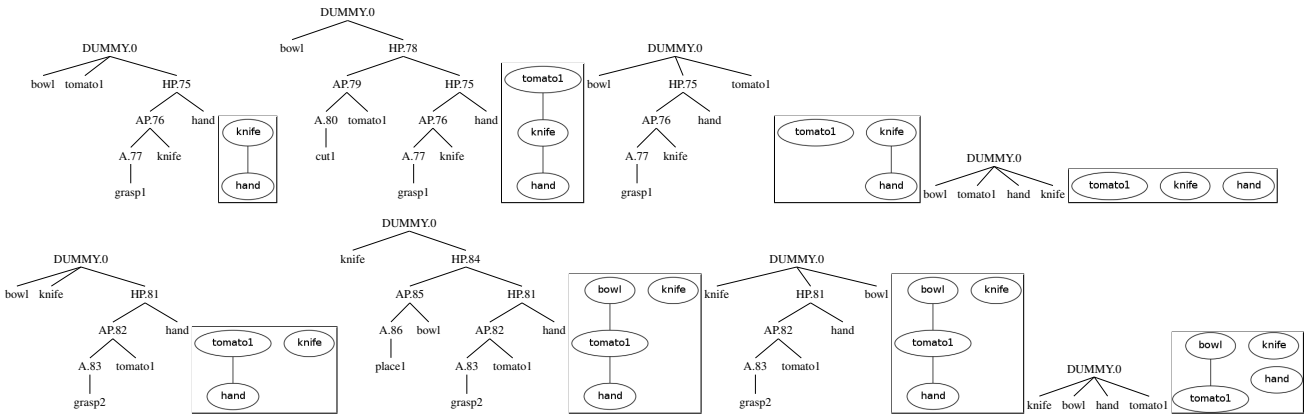
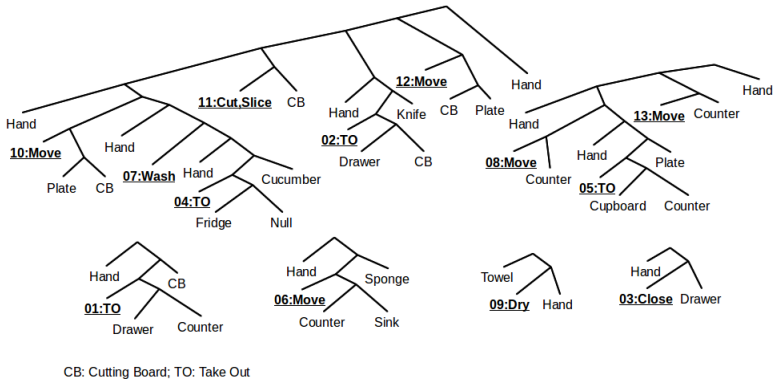


Fig. 2: Example grammatical trees and visual semantic graphs (VSG). The sample sequence is from the 50 Salad dataset.



#Event	Action	Subject	Patient	From/at	To
01	Take Out	Hand	CB	Drawer	Counter
02	Take Out	Hand	Knife	Drawer	CB
03	Close	Hand	Drawer	Null	Null
04	Take Out	Hand	Cucumber	Fridge	Null
05	Take Out	Hand	Plate	Cupboard	Counter
06	Move	Hand	Sponge	Sink	Counter
07	Wash	Hand	Cucumber	Null	Null
08	Move	Hand	Plate	Counter	Null
09	Dry	Towel	hand	Null	Null
10	Move	Hand	Cucumber	Plate	CB
11	Cut Slice	Knife	Cucumber	CB	Null
12	Move	Hand	Cucumber	CB	Plate
13	Move	Hand	Plate	Counter	Null

Fig. 3: Examples of grammatical trees from the TACoS dataset. The original annotation is given in the table.

- inserting a node (between a node and its children)
- deleting a node (and connecting its children to its parent)
- renaming a node (changing the label)

Efficient algorithms exist for finding the minimal edit distance. The algorithm used in this paper is from the original paper by Zhang and Shasha [25]. For the small sizes of trees we encounter, comparing two trees takes negligible time and memory. However, because the number of trees are considerable large, a fast algorithm is desirable. For each pair of manipulation actions (S_n, S_m) $n, m \in (1...N)$ in the 50 Salad dataset, two lists of trees T_n and T_m are parsed in the tree bank. $T_n = \{t_1^n, t_2^n ... t_k^n\}$, $T_m = \{t_1^m, t_2^m ... t_l^m\}$ where k, l are the number of key semantic triplets involved in S_n and S_m respectively. A tree edit distance matrix $D_{(n,m)}$ with a size of $k \times l$ is computed by finding the minimal edit distance between each tree pair T_i^n and T_j^n, $i \in (1...k)$, $j \in (1...l)$,

$$D_{(n,m)} = \begin{pmatrix} D(t_1^n, t_1^m) & D(t_1^n, t_2^m) & \cdots & D(t_1^n, t_l^m) \\ D(t_2^n, t_1^m) & D(t_2^n, t_2^m) & \cdots & D(t_2^n, t_1^m) \\ \vdots & \vdots & & \vdots \\ D(t_k^n, t_1^m) & D(t_k^n, t_2^m) & \cdots & D(t_k^n, t_l^m) \end{pmatrix}. \quad (1)$$

Fig. 4 shows four typical pairs of distance matrices from the 50 salad tree bank. The X and Y axis indicate the temporal order of each sequence respectively.

D. Action Tree Embedding Analysis

The tree bank of manipulation actions can be used to discover the underlying semantic space of the activity. In order to show that the tree bank generated from the manipulation action grammar captures the underlying semantic space of the manipulation action, we apply multidimensional scaling (MDS) on the distance matrix D (which was computed using the minimal edit distance from Sec. V-C.) Specifically, D is constructed by concatenating the $D_{(n,m)}$ for each pair of $n, m \in (1...N)$,

$$D = \begin{pmatrix} D_{1,1} & D_{1,2} & \cdots & D_{1,N} \\ D_{2,1} & D_{2,2} & \cdots & D_{2,N} \\ \vdots & \vdots & & \vdots \\ D_{N,1} & D_{N,2} & \cdots & D_{N,N} \end{pmatrix}. \quad (2)$$

The goal of MDS is, given D, to find I vectors $x_1, \ldots, x_I \in \mathbb{R}^N$ such that $\|x_i - x_j\| \approx D_{i,j}$, for all $i, j \in I$, where $\|\cdot\|$ is a

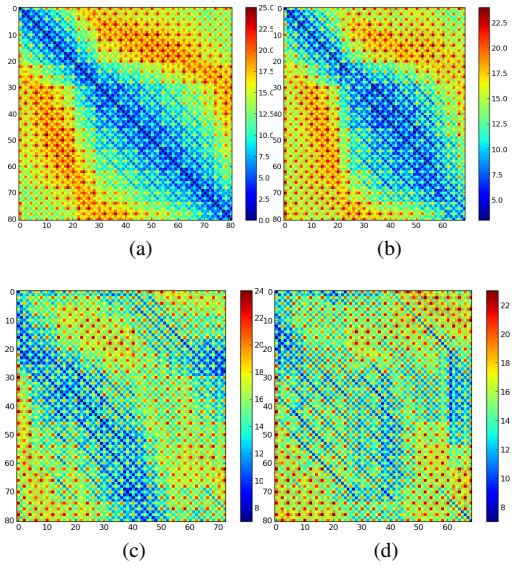


Fig. 4: 50 Salad Dataset, Sequence 3 trail 2. (a) Self distance (b) With Sequence 22 Trail 1, a similar way of making a salad (c) With Sequence 24 Trail 1, making salad in a different order; (d) With Sequence 25 Trail 2, large variation, a total different way of making salad.

vector norm, and I is the total number of action trees in the tree bank. In this work we simply applied the most widely used classical MDS, using the Euclidean distance as norm.

After applying MDS on D from the 50 Salad tree bank, the embedded space is plotted in Fig. 5. Each dot in the figure is a manipulation action tree in the tree bank, and trees from the same manipulation action share the same color. The top three dimensions are used since they best capture the underlying semantic space for the 50 Salad tree bank.

Fig. 5a shows a top-down view of the space. The data points form a ring. This is desirable, because each salad making activity in the dataset starts from an empty tree (no object or tool is used) and ends with an empty tree (both objects and tools are released). Each intermediate step involves the hand(s) using object(s) or tool(s) to apply an action onto another object. The corresponding data points are located on the path of the ring. Fig. 5b shows a side view of the space. The data forms three clusters. This is also desirable, because each intermediate step in salad making involves a **START** step (no object or tool grasped), a **GRASP** step (object or tool grasped) and an **ACT** step (action applied onto object). In fact, from another viewpoint, Fig. 5c shows that the underlying semantic space of salad making in the 50 Salad tree bank indeed consists of three rings.

A trace connecting dots in the space represents each salad making scenario in the 50 salad tree bank. Fig. 5d, 5e, 5f and 5g respectively plot the traces of four typical scenarios discussed in Sec. V-C, which are sequence 3 trail 2, sequence 22 trail 1, sequence 24 trail 1 and sequence 25 trail 2.

E. A Reasoning Example using Action Tree Bank

We organize our tree banks following the Penn tree bank format, which makes it plausible to apply off-the-shelf tools

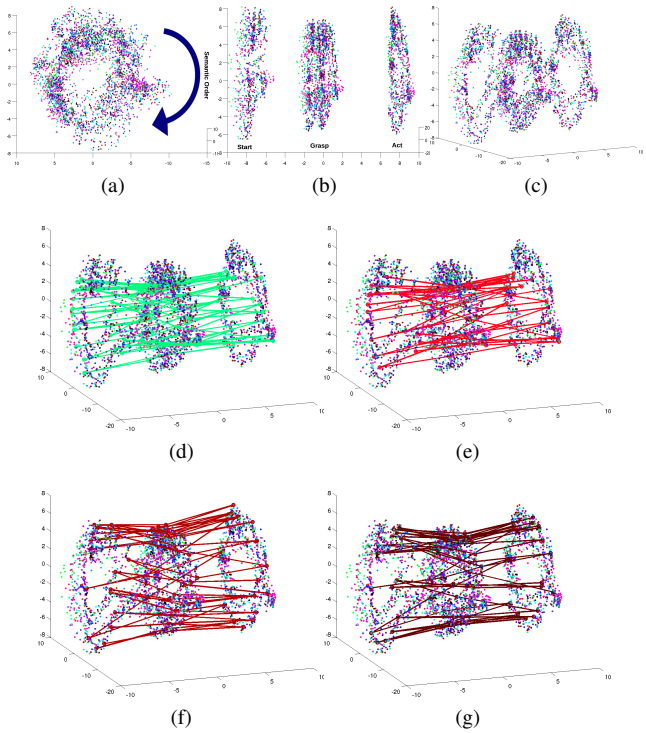


Fig. 5: Manipulation action tree embedding analysis and semantic space discovery.

such as “Tregex” and “tgrep”. These tools are widely used in the computational linguistics community to search tree banks using regular expressions. Thus, humanoid robots that are equipped with the action knowledge in a tree bank fashion have the advantage to do action-centric reasoning easily.

In fig. 6, we show an example using the “Tregex” GUI program to reason in the 50 salad-making tree bank. After observing a “knife” and a “tomato” on the table, the humanoid robot can initialize a search to find sub-trees that contain both “knife” and “tomato” using regular expression. The program returns all the sub-tree matches from the action tree bank. Apparently, in the salad-making scenario, when both “knife” and “tomato” are observed, “hand grasp knife cut tomato” is the most common action to do. This action command can be further used to drive the humanoid. A similar search, which is to find a sub-tree contains both “tomato” and “bowl”, returns an action command “hand grasp tomato place (into) bowl” from the same tree bank. Moreover, since the system is able to return not only the matching sub-trees, but also their temporal locations in each sequences, the robot can easily figure out that “cut tomato” precedes “ tomato place (into) bowl” in this scenario.

In this section we describe one example of using the Penn style tree bank to do reasoning for humanoids. More complicated searches can be conducted by cooperating “and”, “or” or “neglect” logics in the regular expressions. A regular expression generation engine, which can be driven by low-level visual systems such as object recognition and tracking, can generate these searches automatically. Also, it is worth pointing out that there are other more sophisticated tools to

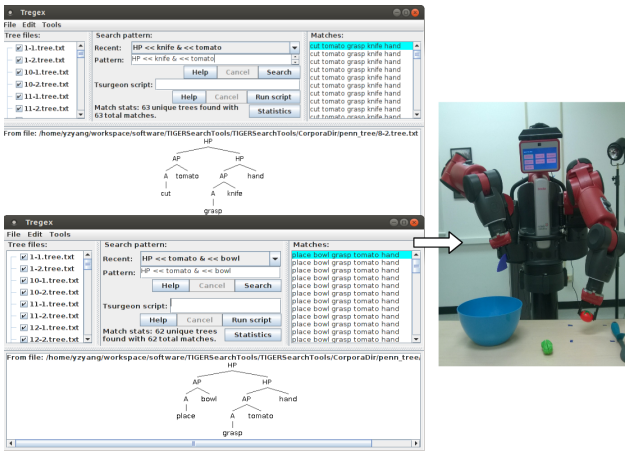


Fig. 6: An example of using "tregex" to achieve action commands from the tree bank knowledge resource.

do reasoning and learning from tree banks. These techniques can also be applied onto autonomous humanoid robots.

VI. Conclusion and Future Works

In this paper, we presented manipulation action tree banks for robot action understanding and execution. We created the knowledge resource using the manipulation action context-free grammar and its associated parsing algorithms. We conducted three experiments on the proposed tree bank. We showed that the tree bank is able to generate visual semantic graphs, and we compared semantic distances to discover the underlying semantic space. Additionally, we showed an example of querying the tree bank to do reasoning for humanoids robot under a specific scenario.

In current work we are integrating the presented linguistic resource with low-level vision tools on a humanoid platform. We are developing a cognitive robot system that equipped with the proposed tree banks deals with the uncertainty as well as the complexity of everyday human manipulation activities. We are also working on an analysis of the different levels of granularity of manipulation tasks in order to possibly take shallow parses of the grammatical trees to find a coarse similarity between different domains of every manipulation actions.

VII. Acknowledgements

This research was funded in part by the support of the European Union under the Cognitive Systems program (project POETICON++), the National Science Foundation under INSPIRE grant SMA 1248056, and support from the US Army, Grant W911NF-14-1-0384 under the Project: Shared Perception, Cognition and Reasoning for Autonomy.

References

[1] W. Yi and D. Ballard, "Recognizing behavior in hand-eye coordination patterns," *International Journal of Humanoid Robotics*, vol. 6, no. 03, pp. 337–359, 2009.

[2] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of english: The penn treebank," *COMPUTATIONAL LINGUISTICS*, vol. 19, no. 2, pp. 313–330, 1993.

[3] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010.

[4] T. Asfour, P. Azad, F. Gyarfas, and R. Dillmann, "Imitation learning of dual-arm manipulation tasks in humanoid robots," *International Journal of Humanoid Robotics*, vol. 5, no. 02, pp. 183–202, 2008.

[5] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, and R. Dillmann, "Toward humanoid manipulation in human-centred environments," *Robotics and Autonomous Systems*, vol. 56, pp. 54–65, 2008.

[6] B. Moldovan, P. Moreno, M. van Otterlo, J. Santos-Victor, and L. De Raedt, "Learning relational affordance models for robots in multi-object manipulation tasks," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4373–4378.

[7] M. R. Dogar, M. C. Koval, A. Tallavajhula, and S. Srinivasa, "Object search by manipulation," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013.

[8] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation: A survey," *Robotics and Autonomous Systems*, 2012.

[9] T. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer vision and image understanding*, vol. 104, no. 2, pp. 90–126, 2006.

[10] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 11, pp. 1473–1488, 2008.

[11] M. Tenorth, J. Ziegltrum, and M. Beetz, "Automated alignment of specifications of everyday manipulation tasks," in *IROS*. IEEE, 2013.

[12] N. Chomsky, *Lectures on government and binding: The Pisa lectures*. Walter de Gruyter, 1993, vol. 9.

[13] K. Pastra and Y. Aloimonos, "The minimalist grammar of action," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 367, no. 1585, pp. 103–117, 2012.

[14] D. Summers-Stay, C. Teo, Y. Yang, C. Fermuller, and Y. Aloimonos, "Using a minimal action grammar for activity understanding in the real world," in *IROS*, 2013.

[15] A. Guha, Y. Yang, C. Fermüller, and Y. Aloimonos, "Minimalist plans for interpreting manipulation actions," *IROS*, 2013.

[16] 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.

[17] S. Lauria, G. Bugmann, T. Kyriacou, J. Bos, and A. Klein, "Training personal robots using natural language instruction," *Intelligent Systems, IEEE*, vol. 16, no. 5, pp. 38–45, Sep 2001.

[18] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions," *Def*, vol. 2, no. 6, p. 4, 2006.

[19] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation." in *AAAI*, 2011.

[20] M. Stenmark and P. Nugues, "Natural language programming of industrial robots," in *Robotics (ISR), 2013 44th International Symposium on*. IEEE, 2013, pp. 1–5.

[21] D. Nyga and M. Beetz, "Everything robots always wanted to know about housework (but were afraid to ask)," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 243–250.

[22] S. Stein and S. J. McKenna, "Combining embedded accelerometers with computer vision for recognizing food preparation activities," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2013)*. ACM, 2013.

[23] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal, "Grounding action descriptions in videos," *Transactions of the Association for Computational Linguistics (TACL)*, vol. 1, pp. 25–36, 2013.

[24] 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.

[25] K. Zhang and D. Shasha, "Simple fast algorithms for the editing distance between trees and related problems," *SIAM journal on computing*, vol. 18, no. 6, pp. 1245–1262, 1989.