Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Robot Learning Manipulation Action Plans by "Watching" Unconstrained Videos from the World Wide Web

Yezhou Yang
University of Maryland
yzyang@cs.umd.edu

Yi Li
NICTA, Australia
yi.li@nicta.com.au

Cornelia Fermüller
University of Maryland
fer@umiacs.umd.edu

Yiannis Aloimonos
University of Maryland
yiannis@cs.umd.edu

Abstract

In order to advance action generation and creation in robots beyond simple learned schemas we need computational tools that allow us to automatically interpret and represent human actions. This paper presents a system that learns manipulation action plans by processing unconstrained videos from the World Wide Web. Its goal is to robustly generate the sequence of atomic actions of seen longer actions in video in order to acquire knowledge for robots. The lower level of the system consists of two convolutional neural network (CNN) based recognition modules, one for classifying the hand grasp type and the other for object recognition. The higher level is a probabilistic manipulation action grammar based parsing module that aims at generating visual sentences for robot manipulation. Experiments conducted on a publicly available unconstrained video dataset show that the system is able to learn manipulation actions by "watching" unconstrained videos with high accuracy.

Introduction

The ability to learn actions from human demonstrations is one of the major challenges for the development of intelligent systems. Particularly, manipulation actions are very challenging, as there is large variation in the way they can be performed and there are many occlusions.

Our ultimate goal is to build a self-learning robot that is able to enrich its knowledge about fine grained manipulation actions by "watching" demo videos. In this work we explicitly model actions that involve different kinds of grasping, and aim at generating a sequence of atomic commands by processing unconstrained videos from the World Wide Web (WWW).

The robotics community has been studying perception and control problems of grasping for decades (Shimoga 1996). Recently, several learning based systems were reported that infer contact points or how to grasp an object from its appearance (Saxena, Driemeyer, and Ng 2008; Lenz, Lee, and Saxena 2014). However, the desired grasping type could be different for the same target object, when used for different action goals. Traditionally, data about the grasp has been acquired using motion capture gloves or hand trackers, such as the model-based tracker of (Oikonomidis, Kyriazis, and Argyros 2011). The acquisition of grasp information from video (without 3D information) is still considered very difficult because of the large variation in appearance and the occlusions of the hand from objects during manipulation.

Our premise is that actions of manipulation are represented at multiple levels of abstraction. At lower levels the symbolic quantities are grounded in perception, and at the high level a grammatical structure represents symbolic information (objects, grasping types, actions). With the recent development of deep neural network approaches, our system integrates a CNN based object recognition and a CNN based grasping type recognition module. The latter recognizes the subject's grasping type directly from image patches.

The grasp type is an essential component in the characterization of manipulation actions. Just from the viewpoint of processing videos, the grasp contains information about the action itself, and it can be used for prediction or as a feature for recognition. It also contains information about the beginning and end of action segments, thus it can be used to segment videos in time. If we are to perform the action with a robot, knowledge about how to grasp the object is necessary so the robot can arrange its effectors. For example, consider a humanoid with one parallel gripper and one vacuum gripper. When a power grasp is desired, the robot should select the vacuum gripper for a stable grasp, but when a precision grasp is desired, the parallel gripper is a better choice. Thus, knowing the grasping type provides information for the robot to plan the configuration of its effectors, or even the type of effector to use.

In order to perform a manipulation action, the robot also needs to learn what tool to grasp and on what object to perform the action. Our system applies CNN based recognition modules to recognize the objects and tools in the video. Then, given the beliefs of the tool and object (from the output of the recognition), our system predicts the most likely action using language, by mining a large corpus using a technique similar to (Yang et al. 2011). Putting everything together, the output from the lower level visual perception system is in the form of (LeftHand GraspType1 Object1 Action RightHand GraspType2 Object2). We will refer to this septet of quantities as *visual sentence*.

At the higher level of representation, we generate a symbolic command sequence. (Yang et al. 2014) proposed a

context-free grammar and related operations to parse manipulation actions. However, their system only processed RGBD data from a controlled lab environment. Furthermore, they did not consider the grasping type in the grammar. This work extends (Yang et al. 2014) by modeling manipulation actions using a probabilistic variant of the context free grammar, and explicitly modeling the grasping type.

Using as input the belief distributions from the CNN based visual perception system, a Viterbi probabilistic parser is used to represent actions in form of a hierarchical and recursive tree structure. This structure innately encodes the order of atomic actions in a sequence, and forms the basic unit of our knowledge representation. By reverse parsing it, our system is able to generate a sequence of atomic commands in predicate form, i.e. as $Action(Subject, Patient)$ plus the temporal information necessary to guide the robot. This information can then be used to control the robot effectors (Argall et al. 2009).

Our contributions are twofold. (1) A convolutional neural network (CNN) based method has been adopted to achieve state-of-the-art performance in grasping type classification and object recognition on unconstrained video data; (2) a system for learning information about human manipulation action has been developed that links lower level visual perception and higher level semantic structures through a probabilistic manipulation action grammar.

Related Works

Most work on learning from demonstrations in robotics has been conducted in fully controlled lab environments (Aksoy et al. 2011). Many of the approaches rely on RGBD sensors (Summers-Stay et al. 2013), motion sensors (Guerra-Filho, Fermüller, and Aloimonos 2005; Li et al. 2010) or specific color markers (Lee et al. 2013). The proposed systems are fragile in real world situations. Also, the amount of data used for learning is usually quite small. It is extremely difficult to learn automatically from data available on the internet, for example from unconstrained cooking videos from Youtube. The main reason is that the large variation in the scenery will not allow traditional feature extraction and learning mechanism to work robustly.

At the high level, a number of studies on robotic manipulation actions have proposed ways on how instructions are stored and analyzed, often as sequences. Work by (Tenorth, Ziegltrum, and Beetz 2013), among others, investigates how to compare sequences in order to reason about manipulation actions using sequence alignment methods, which borrow techniques from informatics. Our paper proposes a more detailed representation of manipulation actions, the *grammar trees*, extending earlier work. Chomsky in (Chomsky 1993) 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 (Pastra and Aloimonos 2012; Summers-Stay et al. 2013; Guha et al. 2013; Yang et al. 2014). (Pastra and Aloimonos 2012) first discussed a Chomskyan grammar for understanding complex actions as a theoretical concept, (Summers-Stay et al. 2013) provided an implementation of such a grammar using as perceptual input only objects. (Yang et al. 2014) proposed a set of context-free grammar rules for manipulation action understanding. However, their system used data collected in a lab environment. Here we process unconstrained data from the internet. In order to deal with the noisy visual data, we extend the manipulation action grammar and adapt the parsing algorithm.

The recent development of deep neural networks based approaches revolutionized visual recognition research. Different from the traditional hand-crafted features (Lowe 2004; Dalal and Triggs 2005), a multi-layer neural network architecture efficiently captures sophisticated hierarchies describing the raw data (Bengio, Courville, and Vincent 2013), which has shown superior performance on standard object recognition benchmarks (Krizhevsky, Sutskever, and Hinton 2013; Ciresan, Meier, and Schmidhuber 2012) while utilizing minimal domain knowledge. The work presented in this paper shows that with the recent developments of deep neural networks in computer vision, it is possible to learn manipulation actions from unconstrained demonstrations using CNN based visual perception.

Our Approach

We developed a system to learn manipulation actions from unconstrained videos. The system takes advantage of: (1) the robustness from CNN based visual processing; (2) the generality of an action grammar based parser. Figure1 shows our integrated approach.

CNN based visual recognition

The system consists of two visual recognition modules, one for classification of grasping types and the other for recognition of objects. In both modules we used convolutional neural networks as classifiers. First, we briefly summarize the basic concepts of Convolutional Neural Networks, and then we present our implementations.

Convolutional Neural Network (CNN) is a multilayer learning framework, which may consist of an input layer, a few convolutional layers and an output layer. The goal of CNN is to learn a hierarchy of feature representations. Response maps in each layer are convolved with a number of filters and further down-sampled by pooling operations. These pooling operations aggregate values in a smaller region by downsampling functions including max, min, and average sampling. The learning in CNN is based on Stochastic Gradient Descent (SGD), which includes two main operations: Forward and BackPropagation. Please refer to (LeCun and Bengio 1998) for details.

We used a seven layer CNN (including the input layer and two perception layers for regression output). The first convolution layer has 32 filters of size 5×5, the second convolution layer has 32 filters of size 5×5, and the third convolution layer has 64 filters of size 5×5, respectively. The first perception layer has 64 regression outputs and the final perception layer has 6 regression outputs. Our system considers 6 grasping type classes.

Grasping Type Recognition A number of grasping taxonomies have been proposed in several areas of research, in-

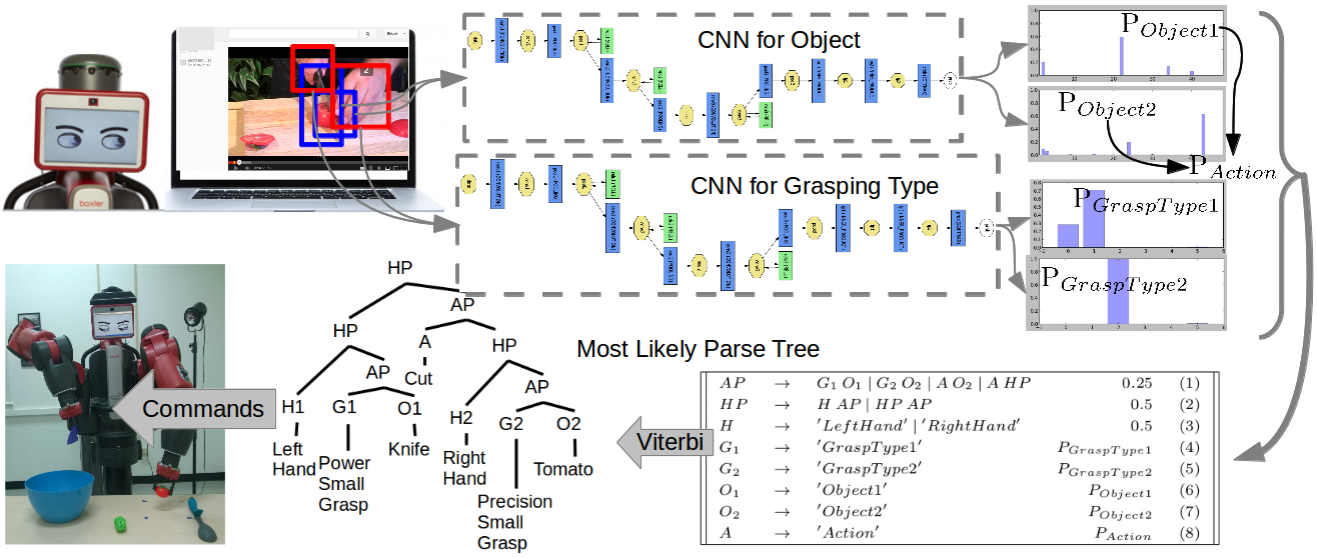


Figure 1: The integrated system reported in this work.

cluding robotics, developmental medicine, and biomechanics, each focusing on different aspects of action. In a recent survey (Feix et al. 2013) reported 45 grasp types in the literature, of which only 33 were found valid. In this work, we use a categorization into six grasping types. First we distinguish, according to the most commonly used classification (based on functionality) into power and precision grasps (Jeannerod 1984). Power grasping is used when the object needs to be held firmly in order to apply force, such as "grasping a knife to cut"; precision grasping is used in order to do fine grain actions that require accuracy, such as "pinch a needle". We then further distinguish among the power grasps, whether they are spherical, or otherwise (usually cylindrical), and we distinguish the latter according to the grasping diameter, into large diameter and small diameter ones. Similarly, we distinguish the precision grasps into large and small diameter ones. Additionally, we also consider a Rest position (no grasping performed). Table 1 illustrates our grasp categories. We denote the list of these six grasps as G in the remainder of the paper.

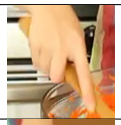
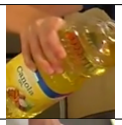
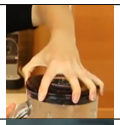
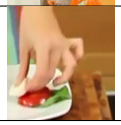
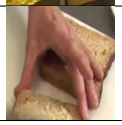
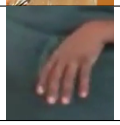
Grasping Types	Small Diameter	Large Diameter	Spherical & Rest
Power			
Precision			

Table 1: The list of the grasping types.

The input to the grasping type recognition module is a gray-scale image patch around the target hand performing the grasping. We resize each patch to 32×32 pixels, and subtract the global mean obtained from the training data.

For each testing video with M frames, we pass the target hand patches (left hand and right hand, if present) frame by frame, and we obtain an output of size $6 \times M$. We sum it up along the temporal dimension and then normalize the output. We use the classification for both hands to obtain (GraspType1) for the left hand, and (GraspType2) for the right hand. For the video of M frames the grasping type recognition system outputs two belief distributions of size 6×1: $P_{GraspType1}$ and $P_{GraspType2}$.

Object Recognition and Corpus Guided Action Prediction The input to the object recognition module is an RGB image patch around the target object. We resize each patch to $32 \times 32 \times 3$ pixels, and we subtract the global mean obtained from the training data.

Similar to the grasping type recognition module, we also used a seven layer CNN. The network structure is the same as before, except that the final perception layer has 48 regression outputs. Our system considers 48 object classes, and we denote this candidate object list as O in the rest of the paper. Table 2 lists the object classes.

apple, blender, bowl, bread, brocolli, brush, butter, carrot, chicken, chocolate, corn, creamcheese, croutons, cucumber, cup, doughnut, egg, fish, flour, fork, hen, jelly, knife, lemon, lettuce, meat, milk, mustard, oil, onion, pan, peanutbutter, pepper, pitcher, plate, pot, salmon, salt, spatula, spoon, spreader, steak, sugar, tomato, tongs, turkey, whisk, yogurt.

Table 2: The list of the objects considered in our system.

For each testing video with M frames, we pass the target object patches frame by frame, and get an output of size $48 \times M$. We sum it up along the temporal dimension and then normalize the output. We classify two objects in the image: (Object1) and (Object2). At the end of classification, the object recognition system outputs two belief distributions of

size 48×1: $P_{Object1}$ and $P_{Object2}$.

We also need the 'Action' that was performed. Due to the large variations in the video, the visual recognition of actions is difficult. Our system bypasses this problem by using a trained language model. The model predicts the most likely verb (Action) associated with the objects (Object1, Object2). In order to do prediction, we need a set of candidate actions V. Here, we consider the top 10 most common actions in cooking scenarios. They are (Cut, Pour, Transfer, Spread, Grip, Stir, Sprinkle, Chop, Peel, Mix). The same technique, used here, was used before on a larger set of candidate actions (Yang et al. 2011).

We compute from the Gigaword corpus (Graff 2003) the probability of a verb occurring, given the detected nouns, $P(Action|Object1, Object2)$. We do this by computing the log-likelihood ratio (Dunning 1993) of trigrams (Object1, Action, Object2), computed from the sentence in the English Gigaword corpus (Graff 2003). This is done by extracting only the words in the corpus that are defined in O and V (including their synonyms). This way we obtain a reduced corpus sequence from which we obtain our target trigrams. The log-likelihood ratios computed for all possible trigrams are then normalized to obtain $P(Action|Object1, Object2)$. For each testing video, we can compute a belief distribution over the candidate action set V of size 10×1 as :

$$P_{Action} = \sum_{Object1 \in O} \sum_{Object2 \in O} P(Action|Object1, Object2) \times P_{Object1} \times P_{Object2}. \quad (1)$$

From Recognitions to Action Trees

The output of our visual system are belief distributions of the object categories, grasping types, and actions. However, they are not sufficient for executing actions. The robot also needs to understand the hierarchical and recursive structure of the action. We argue that grammar trees, similar to those used in linguistics analysis, are a good representation capturing the structure of actions. Therefore we integrate our visual system with a manipulation action grammar based parsing module (Yang et al. 2014). Since the output of our visual system is probabilistic, we extend the grammar to a probabilistic one and apply the Viterbi probabilistic parser to select the parse tree with the highest likelihood among the possible candidates.

Manipulation Action Grammar We made two extensions from the original manipulation grammar (Yang et al. 2014): (i) Since grasping is conceptually different from other actions, and our system employs a CNN based recognition module to extract the model grasping type, we assign an additional nonterminal symbol G to represent the grasp. (ii) To accommodate the probabilistic output from the processing of unconstrained videos, we extend the manipulation action grammar into a probabilistic one.

The design of this grammar is motivated by three observations: (i) Hands are the main driving force in manipulation actions, so a specialized nonterminal symbol H is used for their representation; (ii) an action (A) or a grasping (G) can be applied to an object (O) directly or to a hand phrase (HP), which in turn contains an object (O), as encoded in Rule (1), which builds up an action phrase (AP); (iii) an action phrase (AP) can be combined either with the hand (H) or a hand phrase, as encoded in rule (2), which recursively builds up the hand phrase. The rules discussed in Table 3 form the syntactic rules of the grammar.

AP	$\rightarrow$	$G_1\ O_1 \mid G_2\ O_2 \mid A\ O_2 \mid A\ HP$	0.25	(1)
HP	$\rightarrow$	$H\ AP \mid HP\ AP$	0.5	(2)
H	$\rightarrow$	$'LeftHand' \mid 'RightHand'$	0.5	(3)
G_1	$\rightarrow$	$'GraspType1'$	$P_{GraspType1}$	(4)
G_2	$\rightarrow$	$'GraspType2'$	$P_{GraspType2}$	(5)
O_1	$\rightarrow$	$'Object1'$	$P_{Object1}$	(6)
O_2	$\rightarrow$	$'Object2'$	$P_{Object2}$	(7)
A	$\rightarrow$	$'Action'$	P_{Action}	(8)

Table 3: A Probabilistic Extension of Manipulation Action Context-Free Grammar.

To make the grammar probabilistic, we first treat each sub-rule in rules (1) and (2) equally, and assign equal probability to each sub-rule. With regard to the hand H in rule (3), we only consider a robot with two effectors (arms), and assign equal probability to 'LeftHand' and 'RightHand'. For the terminal rules (4-8), we assign the normalized belief distributions ($P_{Object1}, P_{Object2}, P_{GraspType1}, P_{GraspType2}$,$P_{Action}$) obtained from the visual processes to each candidate object, grasping type and action.

Parsing and tree generation We use a bottom-up variation of the probabilistic context-free grammar parser that uses dynamic programming (best-known as Viterbi parser (Church 1988)) to find the most likely parse for an input visual sentence. The Viterbi parser parses the visual sentence by filling in the most likely constituent table, and the parser uses the grammar introduced in Table 3. For each testing video, our system outputs the most likely parse tree of the specific manipulation action. By reversely parsing the tree structure, the robot could derive an action plan for execution. Figure 3 shows sample output trees, and Table 4 shows the final control commands generated by reverse parsing.

Experiments

The theoretical framework we have presented suggests two hypotheses that deserve empirical tests: (a) the CNN based object recognition module and the grasping type recognition module can robustly recognize input frame patches from unconstrained videos into correct class labels; (b) the integrated system using the Viterbi parser with the probabilistic extension of the manipulation action grammar can generate a sequence of execution commands robustly.

To test the two hypotheses empirically, we need to define a set of performance variables and how they relate to our predicted results. The first hypothesis relates to visual recognition, and we can empirically test it by measuring the precision and recall metrics by comparing the detected object and grasping type labels with the ground truth ones. The second hypothesis relates to execution command generation, and we can also empirically test it by comparing the generated command predicates with the ground truth ones on testing videos. To validate our system, we conducted experi-

ments on an extended version of a publicly available unconstrained cooking video dataset (YouCook) (Das et al. 2013).

Dataset and experimental settings

Cooking is an activity, requiring a variety of manipulation actions, that future service robots most likely need to learn. We conducted our experiments on a publicly available cooking video dataset collected from the WWW and fully labeled, called the Youtube cooking dataset (YouCook) (Das et al. 2013). The data was prepared from 88 open-source Youtube cooking videos with unconstrained third-person view. Frame-by-frame object annotations are provided for 49 out of the 88 videos. These features make it a good empirical testing bed for our hypotheses.

We conducted our experiments using the following protocols: (1) 12 video clips, which contain one typical kitchen action each, are reserved for testing; (2) all other video frames are used for training; (3) we randomly reserve 10% of the training data as validation set for training the CNNs.

For training the grasping type, we extended the dataset by annotating image patches containing hands in the training videos. The image patches were converted to gray-scale and then resized to 32×32 pixels. The training set contains 1525 image patches and was labeled with the six grasping types. We used a GPU based CNN implementation (Jia 2013) to train the neural network, following the structures described above.

For training the object recognition CNN, we first extracted annotated image patches from the labeled training videos, and then resized them to $32 \times 32 \times 3$. We used the same GPU based CNN implementation to train the neural network, following the structures described above.

For localizing hands on the testing data, we first applied the hand detector from (Mittal, Zisserman, and Torr 2011) and picked the top two hand patch proposals (left hand and right hand, if present). For objects, we trained general object detectors from labeled training data using techniques from (Cheng et al. 2014). Furthermore we associated candidate object patches with the left or right hand, respectively depending on which had the smaller Euclidean distance.

Grasping Type and Object Recognition

On the reserved 10% validation data, the grasping type recognition module achieved an average precision of 77% and an average recall of 76%. On the reserved 10% validation data, the object recognition module achieved an average precision of 93%, and an average recall of 93%. Figure 2 shows the confusion matrices for grasping type and object recognition, respectively. From the figure we can see the robustness of the recognition.

The performance of the object and grasping type recognition modules is also reflected in the commands that our system generated from the testing videos. We observed an overall recognition accuracy of 79% on objects, of 91% on grasping types and of 83% on predicted actions (see Table 4). It is worth mentioning that in the generated commands the performance in the recognition of object drops, because some of the objects in the testing sequences do not have

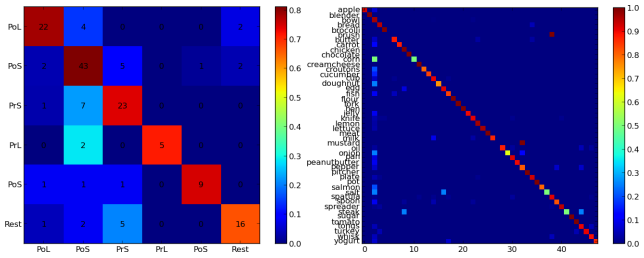


Figure 2: Confusion matrices. Left: grasping type; Right: object.

training data, such as "Tofu". The performance in the classification of grasping type goes up, because we sum up the grasping types belief distributions over the frames, which helps to smooth out wrong labels. The performance metrics reported here empirically support our hypothesis (a).

Visual Sentence Parsing and Commands Generation for Robots

Following the probabilistic action grammar from Table 3, we built upon the implementation of the Viterbi parser from the Natural Language Processing Kit (Bird, Klein, and Loper 2009) to generate the single most likely parse tree from the probabilistic visual sentence input. Figure 3 shows the sample visual processing outputs and final parse trees obtained using our integrated system. Table 4 lists the commands generated by our system on the reserved 12 testing videos, shown together with the ground truth commands. The overall percentage of correct commands is 68%. Note, that we considered a command predicate wrong, if any of the object, grasping type or action was recognized incorrectly. The performance metrics reported here, empirically support our hypothesis (b).

Discussion

The performance metrics reported in the experiment section empirically support our hypotheses that: (1) our system is able to robustly extract visual sentences with high accuracy; (2) our system can learn atomic action commands with few errors compared to the ground-truth commands. We believe this preliminary integrated system raises hope towards a fully intelligent robot for manipulation tasks that can automatically enrich its own knowledge resource by "watching" recordings from the World Wide Web.

Conclusion and Future Work

In this paper we presented an approach to learn manipulation action plans from unconstrained videos for cognitive robots. Two convolutional neural network based recognition modules (for grasping type and objects respectively), as well as a language model for action prediction, compose the lower level of the approach. The probabilistic manipulation action grammar based Viterbi parsing module is at the higher level, and its goal is to generate atomic commands in predicate form. We conducted experiments on a cooking dataset which consists of unconstrained demonstration videos. From the

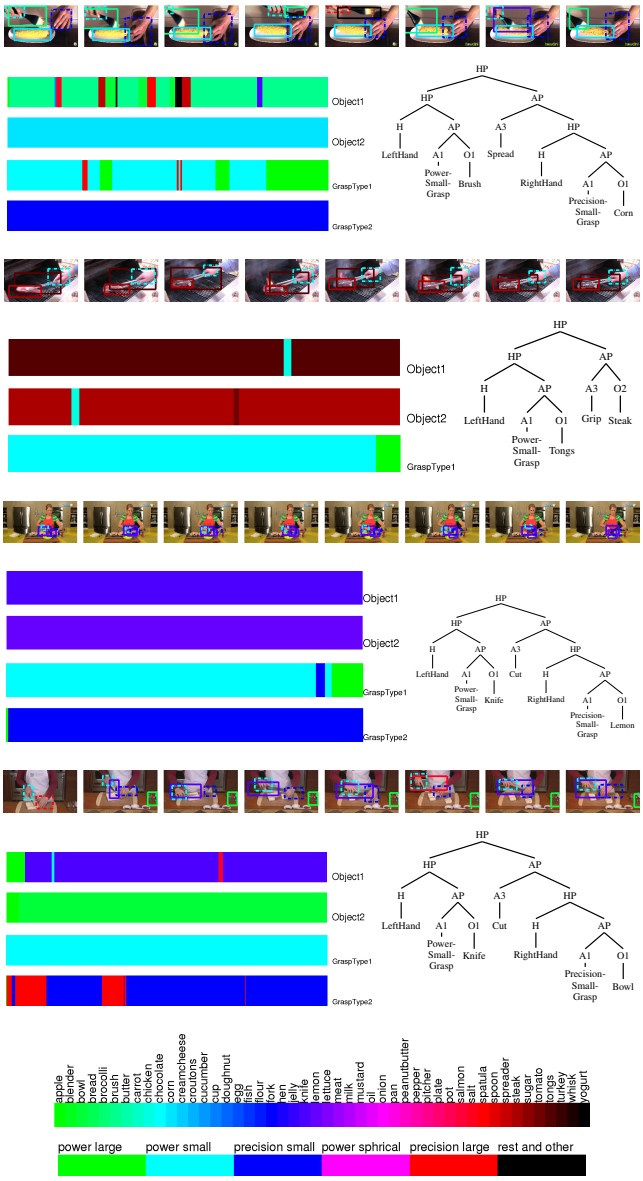


Figure 3: Upper row: input unconstrained video frames; Lower left: color coded (see lengend at the bottom) visual recognition output frame by frame along timeline; Lower right: the most likely parse tree generated for each clip.

performance on this challenging dataset, we can conclude that our system is able to recognize and generate action commands robustly.

We believe that the grasp type is an essential component for fine grain manipulation action analysis. In future work we will (1) further extend the list of grasping types to have a finer categorization; (2) investigate the possibility of using the grasp type as an additional feature for action recognition; (3) automatically segment a long demonstration video into action clips based on the change of grasp type.

Another line of future work lies in the higher level of the system. The probabilistic manipulation action grammar

Snapshot	Ground Truth Commands	Learned Commands
	Grasp_PoS(LH, Knife) Grasp_PrS(RH, Tofu) Action_Cut(Knife, Tofu)	Grasp_PoS(LH, Knife) Grasp_PrS(RH, Bowl) Action_Cut(Knife, Bowl)
	Grasp_PoS(LH, Blender) Grasp_PrL(RH, Bowl) Action_Blend(Blender, Bowl)	Grasp_PoS(LH, Bowl) Grasp_PoL(RH, Bowl) Action_Pour(Bowl, Bowl)
	Grasp_PoS(LH, Tongs) Action_Grip(Tongs, Chicken)	Grasp_PoS(LH, Chicken) Action_Cut(Chicken, Chicken)
	Grasp_PoS(LH, Brush) Grasp_PrS(RH, Corn) Action_Spread(Brush, Corn)	Grasp_PoS(LH, Brush) Grasp_PrS(RH, Corn) Action_Spread(Brush, Corn)
	Grasp_PoS(LH, Tongs) Action_Grip(Tongs, Steak)	Grasp_PoS(LH, Tongs) Action_Grip(Tongs, Steak)
	Grasp_PoS(LH, Spreader) Grasp_PrL(RH, Bread) Action_Spread(Spreader, Bread)	Grasp_PoS(LH, Spreader) Grasp_PrL(RH, Bowl) Action_Spread(Spreader, Bowl)
	Grasp_PoL(LH, Mustard) Grasp_PrS(RH, Bread) Action_Spread(Mustard, Bread)	Grasp_PoL(LH, Mustard) Grasp_PrS(RH, Bread) Action_Spread(Mustard, Bread)
	Grasp_PoS(LH, Spatula) Grasp_PrS(RH, Bowl) Action_Stir(Spatula, Bowl)	Grasp_PoS(LH, Spatula) Grasp_PrS(RH, Bowl) Action_Stir(Spatula, Bowl)
	Grasp_PoL(LH, Pepper) Grasp_PoL(RH, Pepper) Action_Sprinkle(Pepper, Bowl)	Grasp_PoL(LH, Pepper) Grasp_PoL(RH, Pepper) Action_Sprinkle(Pepper, Pepper)
	Grasp_PoS(LH, Knife) Grasp_PrS(RH, Lemon) Action_Cut(Knife, Lemon)	Grasp_PoS(LH, Knife) Grasp_PrS(RH, Lemon) Action_Cut(Knife, Lemon)
	Grasp_PoS(LH, Knife) Grasp_PrS(RH, Broccoli) Action_Cut(Knife, Broccoli)	Grasp_PoS(LH, Knife) Grasp_PoL(RH, Broccoli) Action_Cut(Knife, Broccoli)
	Grasp_PoS(LH, Whisk) Grasp_PrL(RH, Bowl) Action_Stir(Whisk, Bowl)	Grasp_PoS(LH, Whisk) Grasp_PrL(RH, Bowl) Action_Stir(Whisk, Bowl)
Overall Recognition Accuracy	Object: 79% Grasping type: 91% Action: 83%	Overall percentage of correct commands: 68%

Table 4: LH:LeftHand; RH: RightHand; PoS: Power-Small; PoL: Power-Large; PoP: Power-Spherical; PrS: Precision-Small; PrL: Precision-Large. Incorrect entities learned are marked in red.

used in this work is still a syntax grammar. We are currently investigating the possibility of coupling manipulation action grammar rules with semantic rules using lambda expressions, through the formalism of combinatory categorial grammar developed by (Steedman 2002).

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

Aksoy, E.; Abramov, A.; Dörr, J.; Ning, K.; Dellen, B.; and Wörgötter, F. 2011. Learning the semantics of object-action relations by observation. *The International Journal of Robotics Research* 30(10):1229–1249.

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009.

A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57(5):469–483.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35(8):1798–1828.

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural language processing with Python*. " O'Reilly Media, Inc.".

Cheng, M.-M.; Zhang, Z.; Lin, W.-Y.; and Torr, P. H. S. 2014. BING: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*.

Chomsky, N. 1993. *Lectures on government and binding: The Pisa lectures*. Berlin: Walter de Gruyter.

Church, K. W. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, 136–143. Association for Computational Linguistics.

Ciresan, D. C.; Meier, U.; and Schmidhuber, J. 2012. Multi-column deep neural networks for image classification. In *CVPR 2012*.

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, 886–893. IEEE.

Das, P.; Xu, C.; Doell, R. F.; and Corso, J. J. 2013. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Dunning, T. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19(1):61–74.

Feix, T.; Romero, J.; Ek, C. H.; Schmiedmayer, H.; and Kragic, D. 2013. A Metric for Comparing the Anthropomorphic Motion Capability of Artificial Hands. *Robotics, IEEE Transactions on* 29(1):82–93.

Graff, D. 2003. English gigaword. In *Linguistic Data Consortium, Philadelphia, PA*.

Guerra-Filho, G.; Fermüller, C.; and Aloimonos, Y. 2005. Discovering a language for human activity. In *Proceedings of the AAAI 2005 Fall Symposium on Anticipatory Cognitive Embodied Systems*. Washington, DC: AAAI.

Guha, A.; Yang, Y.; Fermüller, C.; and Aloimonos, Y. 2013. Minimalist plans for interpreting manipulation actions. In *Proceedings of the 2013 International Conference on Intelligent Robots and Systems*, 5908–5914. Tokyo: IEEE.

Jeannerod, M. 1984. The timing of natural prehension movements. *Journal of motor behavior* 16(3):235–254.

Jia, Y. 2013. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. 2013. Imagenet classification with deep convolutional neural networks. In *NIPS 2012*.

LeCun, Y., and Bengio, Y. 1998. The handbook of brain theory and neural networks. Cambridge, MA, USA: MIT Press. chapter Convolutional networks for images, speech, and time series, 255–258.

Lee, K.; Su, Y.; Kim, T.-K.; and Demiris, Y. 2013. A syntactic approach to robot imitation learning using probabilistic activity grammars. *Robotics and Autonomous Systems* 61(12):1323–1334.

Lenz, I.; Lee, H.; and Saxena, A. 2014. Deep learning for detecting robotic grasps. *International Journal of Robotics Research* to appear.

Li, Y.; Fermüller, C.; Aloimonos, Y.; and Ji, H. 2010. Learning shift-invariant sparse representation of actions. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2630–2637. San Francisco, CA: IEEE.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60(2):91–110.

Mittal, A.; Zisserman, A.; and Torr, P. H. 2011. Hand detection using multiple proposals. In *BMVC*, 1–11. Citeseer.

Oikonomidis, I.; Kyriazis, N.; and Argyros, A. 2011. Efficient model-based 3D tracking of hand articulations using Kinect. In *Proceedings of the 2011 British Machine Vision Conference*, 1–11. Dundee, UK: BMVA.

Pastra, K., and Aloimonos, Y. 2012. The minimalist grammar of action. *Philosophical Transactions of the Royal Society: Biological Sciences* 367(1585):103–117.

Saxena, A.; Driemeyer, J.; and Ng, A. Y. 2008. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research* 27(2):157–173.

Shimoga, K. B. 1996. Robot grasp synthesis algorithms: A survey. *The International Journal of Robotics Research* 15(3):230–266.

Steedman, M. 2002. Plans, affordances, and combinatory grammar. *Linguistics and Philosophy* 25(5-6):723–753.

Summers-Stay, D.; Teo, C.; Yang, Y.; Fermüller, C.; and Aloimonos, Y. 2013. Using a minimal action grammar for activity understanding in the real world. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4104–4111. Vilamoura, Portugal: IEEE.

Tenorth, M.; Ziegltrum, J.; and Beetz, M. 2013. Automated alignment of specifications of everyday manipulation tasks. In *IROS*. IEEE.

Yang, Y.; Teo, C. L.; Daumé III, H.; and Aloimonos, Y. 2011. Corpus-guided sentence generation of natural images. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 444–454. Association for Computational Linguistics.

Yang, Y.; Guha, A.; Fermuller, C.; and Aloimonos, Y. 2014. A cognitive system for understanding human manipulation actions. *Advances in Cognitive Sysytems* 3:67–86.