Towards Incremental Learning of Task-dependent Action Sequences using Probabilistic Parsing

Kyuhwa Lee and Yiannis Demiris
Department of Electrical and Electronic Engineering
Imperial College London, SW7 2BT, UK
Email: {k.lee09, y.demiris}@imperial.ac.uk

***Abstract*—We study an incremental process of learning where a set of generic basic actions are used to learn higher-level task-dependent action sequences. A task-dependent action sequence is learned by associating the goal given by a human demonstrator with the task-independent, general-purpose actions in the action repertoire. This process of contextualization is done using probabilistic parsing. We propose stochastic context-free grammars as the representational framework due to its robustness to noise, structural flexibility, and easiness on defining task-independent actions. We demonstrate our implementation on a real-world scenario using a humanoid robot and report implementation issues we had.**

I. Introduction

There has been a growing interest in developing autonomous robots which are capable of learning goal-directed actions by imitating humans using multi-level representations of actions [1][2][3]. Broadly speaking, there are two main benefits of enabling a robot to learn a new behavior by imitation.

From an engineering perspective, imitation learning provides a means to speed up learning a new behavior without exhaustive manual programming. This will give people not familiar with robot programming the ability to teach robots to perform tasks.

From a scientific perspective, on the other hand, as the robotics domain blends engineering with psychology and neuroscience, it is recognized as a new tool to investigate cognitive and biological questions, as discussed by Schaal [4] and Demiris [5]. Learning algorithms which can be implemented on robotic platforms illuminate gaps between theories and real world, and allows research to focus on filling these gaps. They also provide a means to predict the expected result, which might be an important tool for directing further experiments. [6].

In the real-world environment, there are still many obstacles yet to be solved for a robot to be successful on imitation learning. One of them is dealing with low-level complexities on vision-based robotic systems in real-world environment, such as noise and occlusions. It is often preferable to minimize low-level errors to allocate more resource on solving higher-level problems. As an analogy, suppose that a man is trying to lift a cup. Even though *grasping* a handle is only partially visible or not visible at all due to its subtle finger manipulations, we can still assume that he grasped it by observing the cup being lifted in the air without paying significant amount of attention on fingers. Our motivation comes from the realization that if a robot has knowledge about a minimal set of basic actions which are frequently used in human-robot interaction environment, it can boost the performance of learning new concepts using these basic "vocabularies".

Formally, our problem falls into the domain of "what to imitate", among five fundamental categories on imitation learning suggested by Dautenhahn and Nehaniv [7]. As discussed in [8][9], the question of "what to imitate" primarily deals with understanding the goal or intention of the demonstrator. In our experiment, we represent the actions in terms of goals instead of action trajectories. This is also partially rooted on the experiments of Baldwin and Woodward, which show that humans even from a very early age tend to interpret actions based on goals rather than motion trajectories [10][11].

We use the hand as a reference cue that describes the observation. Flanagan and Johansson [12] elegantly demonstrated in their experiments, where participants watched a series of block-moving tasks, that people tend to map the visual representation of the observed action onto a motor representation of the same action, instead of a purely visual analysis of the elements independent from actuators. In both our and Flanagan's cases, hand is equivalent to the actuator which forms the basis of the visual representation of objects.

In this paper, we make the following contributions.

1) We present a prototypical incremental learning approach which contextualizes task-independent generic action sequences into task-dependent action sequences.

2) We validate our implementation on 94 samples obtained from human participants to investigate possible benefits and limitations in a real-world environment.

II. Incremental Learning Process

A. Approach

Our goal is to make the system learn actions that are task-specific by observing human demonstrations given a set of task-independent general-purpose action set. Our method is divided into two stages. In the first stage, we train the system with a set of basic actions that could be re-used in multiple domains. The choice of learning technique is up to the system designer's decision, although sequential models such as Hidden Markov Models (HMMs) or Conditional Random Fields (CRFs) are often used. [13] We employed HMMs for our experiment. In the second stage, we define higher-level task-independent actions that are composed of rudimentary actions learned in the first stage as basic vocabularies. In our work, we use stochastic context-free grammars to represent these actions. Figure 1 shows the two stages of our learning process.

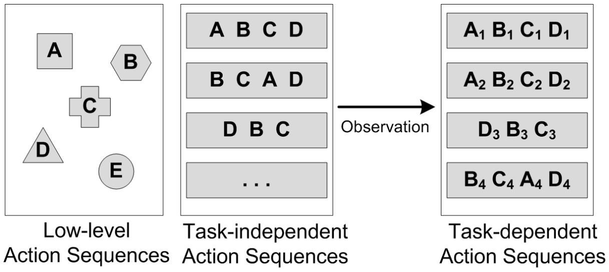

Fig. 1. Building task-dependent actions by associating goals with task-independent actions. Domain-independent low-level action sequences are learned in the first stage to be used as basic vocabularies for representing higher-level task-independent actions. By observing a human demonstration and parsing the observation, the system classifies the corresponding actions and assigns the goal label, e.g. 1, 2, 3 or 4.

Using stochastic context-free grammars (SCFGs) to represent higher-level actions provides strong benefits on recognizing human activities. For example, Moore et al. [14] applied SCFG to represent and recognize various actions used in Blackjack game, which showed good capability to deal with errors. In [15], Ivanov et al. applied SCFG to recognize conducting gestures by extending the original parsing algorithm to consider input symbols with uncertainty (probability) values which resulted in robust recognition. In our case, however, we bring this approach to the robotic learning domain to construct general models of human behaviors (task-independent actions) that can be re-used for various kinds of tasks. The input to the SCFG parser is the sequence of symbolized low-level actions, which are the output of HMMs in our case.

Although providing task-independent actions in prior might seem heuristic to some extent, we posit these mid-level representations are crucial for efficient interactions between humans and robots instead of learning from the scratch. In the following section, we consider a sample scenario to discuss about the possible ways to realize the aforementioned functionalities.

B. Test Scenario

In this section, we illustrate a sample real-world scenario and discuss about possible ways of implementation. Consider a scenario where we want to organize various types of objects using a box. Depending on object type, each should be treated differently: there could be objects that could be simply dropped into the box whereas some fragile objects might need to be placed safely inside the box. Also, if the object does not fit into the box, it should be placed next to it. The representations of these three handling methods are given to the robot in the form of stochastic context-free grammars (SCFG) but the robot has no prior information about objects. Based on this scenario, we want to teach a robot how each object should be handled by demonstrating proper action sequences for each object.

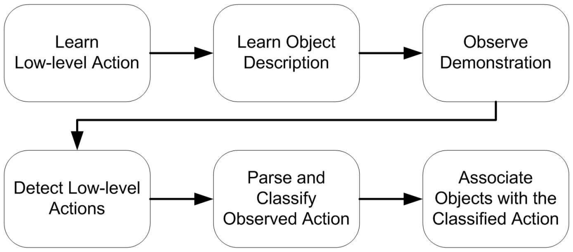

Fig. 2. Overview of the implementation for our test scenario.

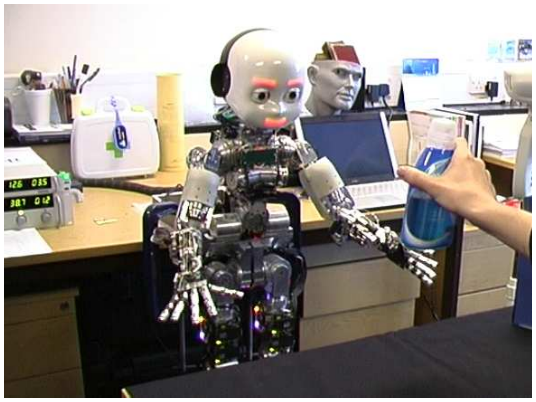

Fig. 3. iCub, the humanoid robot used in the experiments observing object-specific handling sequences by human demonstration.

To recognize these action sequences, it is a natural requirement that the system should be able to recognize the meaningful low-level actions such as a) approach or leave away from the object, b) grasp or release the object, c) move the object closer to the box. The models to recognize these action components are learned in prior. Further details about these actions can be found in Section III-C.

III. Implementation

A. Overall Process

Based on the scenario illustrated in the previous section, we implement our approach as described in Figure 2. The system first learns the demonstrator's skin color histogram by extracting a patch from the detected face and uses it to track the demonstrator's hand. It subsequently learns the color histogram of the object chosen by the demonstrator in the form similar to that on Figure 3. Further details about trackers will be discussed in Section III-B.

It subsequently observes the demonstrator performing an action sequence and generates a series of low-level action symbols using learned models. Detailed method regarding recognizing low-level actions will be discussed in more detail in Section III-C.

The symbols generated in the last step are fed into a SCFG parser to classify the action that the demonstrator has performed, e.g. `Place the object in the box`. The object description (learned color histogram) is associated with the classified action, e.g. `Place the` *`blue`* `object in the box`. This process will be discussed in Section III-D.

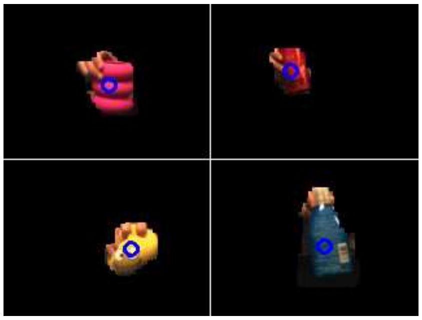

Fig. 4. Examples of object segmentation.

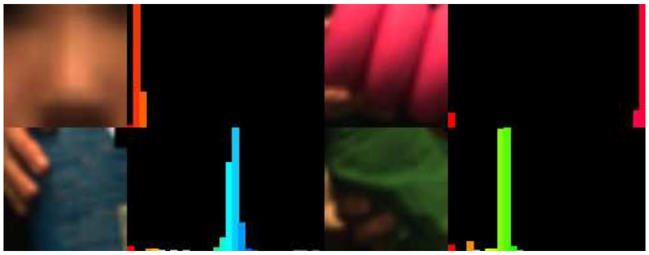

Fig. 5. Extracted patches and their color histograms. In histogram images, x-axis represents the color bin and y-axis represents the frequency. Finger colors in the patch are compensated for better tracking performance.

B. Trackers

We use hand and object trackers based on the CamShift tracking algorithm implemented in [16]. Hand color histograms are learned from the face patch of the demonstrator in the beginning, and used throughout the experiment until all action sequences have been performed. Object patches are obtained when a user holds an object close to the system where its distance is measured from depth perception using stereo camera. The system learns the object color histogram before observing each action.

The method we used allows the system to learn an object in a natural way from humans with high success rate. It worked as expected on most of trials although there were occasionally flickering noise on the border area. In our experiments, wrong object patches were learned only 4 times out of 100 trials. An example object segmentation can be seen on Figure 4. We average the positions of each tracker every 3 frames and use them as input to the low-level detectors to increase the tracker stability.

C. Low-level Action Detectors

Low-level detectors compute the probability of certain types of events being occurred from pixel-level data. Examples include low-level motions such as `approaching an object` and object states such as `object observable`. As long as they provide the probability or confidence values between 0 and 1, any low-level detectors can be used, e.g. aural or tactile event detectors. The output values coupled with probability, or certainty, are called terminals. The systems uses 7 event detectors in total, as described below. We denote H for hand, O for object, and B for box.

1) 'H approaching O', 'H leaving away from O', 'O approaching B', 'O leaving away from B': They represent the relationships between two entities. The system learned two general types of HMM, 'Approaching' and 'Moving away from' offline using 20 tracked video samples. The input to each HMM is the sign change of distance between two entities, i.e. {-,+,0}. The HMM library of [17] was used.

2) 'Object visibility' and 'Hand visibility': These two symbols represent the observability of objects. Probabilities are obtained by computing the Bhattacharyya distance between the histogram of the current object tracking window and its previously learned histogram. Color bin size of 32 is used for the experiment. The above function outputs the histogram distance between 0 and 1, where 0 means two histograms are identical. Ideally, if an object is placed in a box, its visibility should reach 1.

3) 'In contact with object': This detector is a Gaussian function with parameters learned from 50 samples of distances between hand and object center positions while holding an object.

D. Action Parsing

From the input stream of terminals generated by low-level detectors discussed in Section III-C, we need to find the action sequence from the action sequence repertoire that best explains the observation. Stochastic Context-Free Grammars (SCFGs) are well suited for this purpose due to its robustness against noise and easiness on defining actions. Advantages on using SCFG model on imitation learning are as follows:

First, it can utilize syntactic knowledge instead of relying on pure statistics to solve a problem as they can be expressed using mid-level representations, e.g. "drop an object". Second, it can disambiguate the noisy actions at the low level using the parsed result. Once the parsing is finished, the action grammar rule with the highest probability is selected and used to explain the input symbols generated by the low-level detectors. Third, although it shares many properties with HMM, it inherently supports more general models, e.g. counting models such as $a^n b^n$. Last but not least, because of its compact representation using linguistic constructs, it allows a wide range of users to define actions which does not require high level of technical skills.

An action is defined using terminals, non-terminals and rule probabilities. A terminal, conventionally written in lower case, is generated by a low-level detector with an associated probability. It can be easily added by defining an additional event detector. A non-terminal, conventionally written in upper case, is an intermediate symbol that can be regarded as a higher-level description. Rule probability, similar to transition probability in HMM, is applied when the state is expanded.

A stochastic context-free grammar (SCFG) parser receives input a sequence of N dimensional vectors where N is the number of terminals. It then parses them to find the most probable rule that best explains the observation and outputs probabilities of each possible action. SCFG is essentially a

TABLE I
ACTION GRAMMAR OF DROP

BEGIN	⇒	DROP		[1.0]
DROP	⇒	AOBJ CONTACT ABOX LOBJ OGONE		[1.0]
AOBJ	⇒	AOBJ	aobj	[0.5]
	|	aobj		[0.4]
	|	SKIP	aobj	[0.1]
ABOX	⇒	ABOX	abox	[0.5]
	|	abox		[0.4]
	|	SKIP	abox	[0.1]
CONTACT	⇒	CONTACT	contact	[0.5]
	|	contact		[0.4]
	|	SKIP	contact	[0.1]
LOBJ	⇒	LOBJ	lobj	[0.5]
	|	lobj		[0.4]
	|	SKIP	lobj	[0.1]
OGONE	⇒	OGONE	ogone	[0.5]
	|	ogone		[0.4]
	|	SKIP	ogone	[0.1]

* Naming conventions: OBJ=object, BOX=box, A=approach, L=leave
HGONE=hand visibility, OGONE=object visibility
CONTACT=hand in contact with an object, SKIP=See Section III-D

stochastic model that extends context-free grammar similar to HMM which extends regular grammars.

As an example, definition of action DROP used in the experiment is shown in Table I. When we see the rule that expands "AOBJ", there are three possibilities that could be interpreted, with probability 0.5, 0.4 and 0.1, respectively. If one wants to incorporate a top-down knowledge on a specific action, it can be realized by biasing the rule probabilities.

The "SKIP" symbol can be thought of as a wildcard which can accept any symbol. It gives "tolerence" to noise symbols that are out of context and it is usually set to low probability. If the low-level detectors generate too much noise, the overall parsed result gets lower probability(confidence).

The terminals are given as input in the form of vector, of which each element is represented with probability. For each position of the input stream, the parser keeps a set of states which represents all the pending derivations. Since the state transition is occurred in non-deterministic way, a large number of pending derivations can be generated.

We briefly explain about the parser using some of the terminology used in conventional context-free grammar model. The parser begins from the start state and iterates over three basic steps: *scanning, completion, and prediction*. For detailed description, please refer to [15].

On *Scanning* step, a symbol is read from the input and matched against all pending states, starting from the start state. The rules which do not comply with the observation are rejected and the corresponding derivations are pruned from the parse tree.

On *Completion* step, given a set of productions which have been confirmed on the *Scanning* step, the parser advances the current positions in the parse tree.

On *Prediction* step, the parser hypothesizes the prospective input based on current position in the parse tree. It adds the next possible state from the current position to the list of pending states to be confirmed on the *Scanning* step.

These three steps are iterated until the end of input stream or it satisfies the stop condition, e.g. end of demonstration. A Viterbi path is computed during parsing as a single derivation path with the maximum path probability.

As discussed in [18], the time complexity of Earley's parser is $O(l^3)$, where l is the length of symbols. It decreases to $O(l^2)$ if a grammar is unambiguous, i.e. the number of distinct derivation trees of a sentence is 1.

IV. EXPERIMENTS

Based on the scenario described in Section II-B and the implementation described in Section III, we conducted our experiments with 10 participants repeating 10 demonstrations each. In this experiment, we use a humanoid robot, iCub, as shown on Figure 3. iCub is a child-sized humanoid robot with 53 degrees of freedom. It is equipped with PointGrey DragonFly II cameras for both eyes.

In this experiment, human demonstrators choose any object from the selection of sponge dolls, a ceramic doll, two types of fruits and a water bottle, and perform one of three object handling actions described in Section III. Namely, these actions are NEXTBOX (place the object next to the box), PLACE (place the object inside the box), and DROP (drop the object into the box). The choice of an object and corresponding action sequence is fully up to the demonstrator's will. There is also no restriction on the demonstrator's performing speed and movement trajectories as long as they think it is meaningful.

The participant sits on a chair approximately 1.2m distant from the robot and a table is placed in the middle. The participant is allowed to sit a little bit closer or farther from the robot if it felt more comfortable. The participant starts experiment by showing an object to the robot and performing an action in mind.

After the demonstrator has finished performing actions, the iCub confirms the result by pointing to each object and showing corresponding actions using gestures. The reason it shows gestures instead of actually manipulating the objects is solely because of grasping strength issues with our iCub model.

Part of the grammar rules that relates to the action DROP is shown on Table I. Non-terminal symbols from AOBJ to OGONE are added only to handle repetitive symbols and erroneous symbols. In our case, the probability of entering SKIP rule is set to 0.1 based on heuristics.

After learning a series of 3 actions, the demonstrator places 3 objects used in the experiment in front of the iCub, which then performs to explain what it has learned. Since its inverse kinematics module is not accurate enough to grab and hold an object, it instead performs a grabbing gesture after pointing to an object and execute the remaining part, such as releasing its hand on the side of the box(NEXTBOX), above the box(DROP),

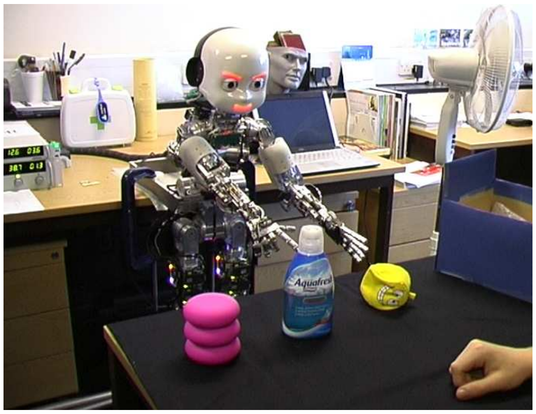

Fig. 6. iCub imitating learned sequence of actions. It points to each object and performs grabbing gesture accordingly, followed by appropriate arm movements depending on the recognized result.

TABLE II
SAMPLE PROBABILISTIC SYMBOLS GENERATED BY LOW-LEVEL DETECTORS AND THE PARSED RESULT

time	abox	lbox	aobj	lobj	contact	ogone	hgone
1	0.1174	0.1426	0.6868	0.0532	0.0000	0.1800	0.1985
2	0.5284	0.0136	0.0336	0.4245	0.0000	0.3826	0.1726
3	0.4796	0.0216	0.0512	0.4476	0.0000	0.3627	0.2095
4	0.2098	0.0640	0.6849	0.0413	0.0000	0.3103	0.2053
5	0.1590	0.0681	0.7359	0.0370	0.0000	0.3186	0.3366
6	0.1598	0.0654	0.7477	0.0270	0.0001	0.1427	0.5125
7	0.1208	0.0930	0.7614	0.0248	0.0013	0.2728	0.5846
8	0.3048	0.0277	0.6464	0.0210	1.0000	0.2159	0.6022
9	0.3261	0.0254	0.6296	0.0189	1.0000	0.1977	0.6196
10	0.2905	0.2511	0.1193	0.3392	0.0000	0.8438	0.2689
11	0.3092	0.2697	0.1366	0.2846	0.0000	0.8446	0.2708
12	0.4722	0.4753	0.0328	0.0197	0.0000	0.8549	0.2335

(Numbers are rounded on the fourth digit after decimal point.)

>> Action sequences recognized as DROP.
>> Parsed action sequences:
aobj aobj aobj aobj aobj aobj aobj contact abox lobj ogone ogone

TABLE III
RESULTS. N:NEXTBOX, D:DROP, P:PLACE, X:RECOGNITION FAILURE

Gt \ Ob	**N**	**D**	**P**	**X**	**Sum**
N	85	7	0	2	94
D	8	76	7	3	94
P	7	22	60	5	94
Sum	100	105	67	10	282

TABLE IV
CONFUSION MATRIX

Gt \ Ob	**N**	**D**	**P**	**X**
N	0.90	0.07	0.00	0.02
D	0.09	0.81	0.07	0.03
P	0.07	0.23	0.64	0.05

or on the back of the box(PLACE). An example of the iCub explaining to the demonstrator is shown on Figure 6.

V. RESULTS AND ANALYSIS

A total of 100 sets of experiments were performed, excluding 6 sets that were not usable due to recording problems. Typical single action demonstration spans between 2 and 6 seconds. In some extreme cases, actions were extremely fast (less than 1 second) or slow (more than 20 seconds). In this experiment, only the performance of recognizing actions is evaluated, not object recognition, as the latter belongs to another problem domain.

Table II shows an example output of low-level detectors and the parsed result obtained by the stochastic parser. The terminal symbols in the last line denotes the most probable terminal path reached based on the overall observation. Tables III and IV shows the raw scores and confusion matrix, respectively.

It is worth noting that “aobj” (approach object) symbol has low probability on time steps 2 and 3 (0.0336 and 0.0512, respectively) which is supposed to be high as “DROP” action expects to observe only “aobj” symbols until grabbing the object. However, after the whole action is recognized as “DROP”, these ambiguous symbols are parsed correctly as “aobj”. It will enable the learner to perform the exact timings of actions, e.g. it knows when to stop approaching the object, when to touch the object, and when to approach the box. Although timing is not critical in our example, one could easily imagine other kinds of tasks where it is more important, e.g. playing musical instruments.

Table III shows the actual number of trials and errors made in this experiment. *Gt* denotes *Ground truth* while *Ob* denotes *Observed result*. X denotes the case where the algorithm fails to find the answer due to extremely low probabilities. It occurs if there are too many symbols that are inconsistent with all of the defined rules. It generally happens more often on lengthy demonstrations.

Table IV shows the confusion matrix of the overall result. The accuracy of the NEXTBOX action is high because it is fairly easy to recognize the action due to its simple structure. The PLACE actions were recognized as DROP in more than 20% of the trials. This is mainly due to the error made on the position of the tracker window or significantly different lighting conditions, such as reflection.

If the demonstration is done too slowly, the tracker often suffers from “jitter” effect which increases the error on the output. This problem could be alleviated by applying Kalman filter on the tracker but we have not used it in this work. As can be seen on time steps 2 and 3 in Table II, even when the hand was approaching the object, low-level detector occasionally recognized “approaching” as low probability and “leaving” as hi1gh probability.

VI. DISCUSSIONS AND FUTURE DIRECTIONS

It is possible to learn the structure and probabilities of rules, but it is commonly regarded as intractable, as discussed in [19]. However, as Lari and Young discussed in [20], it is not impossible to estimate the probabilities once the structure is fixed using an *inside-outside algorithm*. In our implementation, these grammar rules were given manually as we are concerned on using SCFGs to represent mid-level actions. For studies on inferring structures and parameters, we direct readers to [21] and [22].

A current limitation is that it is necessary to know when to start and stop observing. It is possible to work-around this problem by adding vocal commands or specific gestures made by the demonstrator, but they are essentially still equivalent to manual manipulation.

There are some interesting topics on extending this work. First, by understanding the goal or intention of the action, it is possible to direct the robot's attention towards a more informative spot. It will help the robot to obtain more context-related information that could boost the perception performance, as discussed in [23]. This is essentially a top-down approach on directing the robot's attention.

Another interesting research topic is to enhance the parser by using the *state* information. Currently, the terminals are generated based on events. Hence, it is not suitable to represent simultaneous actions, e.g. `holding an object` *`while`* `approaching a box`. By integrating the notion of *state*, it is possible to describe wider range of actions more effectively.

Finally, it is also possible to extend the framework to take advantage of multi-sensory input such as sound or tactile sensing by incorporating additional low-level detectors.

VII. Conclusions

We have presented a prototype of an incremental learning method using two levels of action hierarchies. We trained low-level action models and used them to generate symbols which are used to parse higher-level actions. These classified action sequences are associated with object descriptions so that the robot could perform the correct action sequence later when the similar object is found. We have shown that this could be done effectively by incorporating mid-level representations of actions using lower-level primary action models as building blocks. Any low-level detectors can be designed that best suit the situation which makes this method scalable.

Another advantage of the proposed method is that each mid-level action set can be defined using conventional grammar-like style which is intuitive and human-readable. This allows a wider range of users to take full advantage of it.

Finally, we have evaluated our method in a real-world setting using the iCub to find out the possible advantages and drawbacks.

We anticipate that this work will contribute to the communities of imitation learning and related areas. The system has potential applications in human-robot interaction studies, where recognizing and contextualizing human actions is necessary.

Acknowledgments

Thanks to Balint Takacs, Aaron Bobick, Yan Wu, Harold Soh, Raquel Ros Espinoza and many others for their help on this work. This research was partially supported by the EU FP7 project EFAA (FP7-ICT-270490).

References

[1] M. Pardowitz, S. Knoop, R. Dillmann, and R. Zollner, "Incremental learning of tasks from user demonstrations, past experiences, and vocal comments," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 2, pp. 322–332, 2007.

[2] B. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, pp. 469–483, 2009.

[3] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, pp. 799–822, 1994.

[4] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[5] J. Demiris and G. Hayes, "Imitation as a dual-route process featuring predictive and learning components; a biologically plausible computational model," *Imitation in animals and artifacts*, p. 327, 2002.

[6] Y. Demiris and A. Meltzoff, "The robot in the crib: A developmental analysis of imitation skills in infants and robots," *Infant and Child Development*, vol. 17, no. 1, pp. 43–53, 2008.

[7] K. Dautenhahn and C. Nehaniv, "The agent-based perspective on imitation," *Imitation in animals and artifacts*, p. 1, 2002.

[8] A. Lockerd and C. Breazeal, "Tutelage and socially guided robot learning," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings*, vol. 4, 2004.

[9] S. Calinon, F. Guenter, and A. Billard, "Goal-directed imitation in a humanoid robot," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005*, 2005, pp. 299–304.

[10] J. Baird and D. Baldwin, "Making sense of human behavior: Action parsing and intentional inference," in *Intentions and Intentionality*, F. Malle, L. Moses, and D. Baldwin, Eds. Cambridge, MA: MIT Press, 2001, ch. 9.

[11] A. Woodward, J. Sommerville, and J. Guajardo, "How infants make sense of intentional action," *Intentions and intentionality: Foundations of social cognition*, pp. 149–169, 2001.

[12] J. Flanagan and R. Johansson, "Action plans used in action observation," *Nature*, vol. 424, no. 6950, pp. 769–771, 2003.

[13] M. Ahad, J. Tan, H. Kim, and S. Ishikawa, "Human activity recognition: various paradigms," in *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*. IEEE, 2008, pp. 1896–1901.

[14] D. Moore and I. Essa, "Recognizing multitasked activities from video using stochastic context-free grammar," in *Proceedings of the National Conference on Artificial Intelligence*. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002, pp. 770–776.

[15] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 852–872, 2000.

[16] G. Bradski, "The opencv library," *Doctor Dobbs Journal*, vol. 25, no. 11, pp. 120–126, 2000.

[17] D. Lin. (2003) A c++ implementation of hidden markov model.

[18] A. Stolcke, "An efficient probabilistic context-free parsing algorithm that computes prefix probabilities," in *Computational Linguistics, MIT Press for the Association for Computational Linguistics*, 1995, vol. 21.

[19] C. de la Higuera, "A bibliographical study of grammatical inference," *Pattern Recognition*, vol. 38, no. 9, pp. 1332–1348, 2005.

[20] K. Lari and S. Young, "The estimation of stochastic context-free grammars using the inside-outside algorithm," *Computer speech and Language*, vol. 4, no. 1, pp. 35–56, 1990.

[21] K. Kitani, S. Yoichi, and A. Sugimoto, "Recovering the basic structure of human activities from noisy video-based symbol strings," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22 Issue 8, pp. 1621–1646, 2008.

[22] A. Stolcke and S. Omohundro, "Inducing probabilistic grammars by bayesian model merging," *Grammatical Inference and Applications*, pp. 106–118, 1994.

[23] Y. Demiris and B. Khadhouri, "Content-based control of goal-directed attention during human action perception," *Interaction Studies*, vol. 9, no. 2, pp. 353–376, 2008.