Automatic Segmentation and Recognition of Human Activities from Observation based on Semantic Reasoning

Karinne Ramirez-Amaro[1], Michael Beetz[2] and Gordon Cheng[1]

***Abstract*— Automatically segmenting and recognizing human activities from observations typically requires a very complex and sophisticated perception algorithm. Such systems would be unlikely implemented *on-line* into a physical system, such as a robot, due to the pre-processing step(s) that those vision systems usually demand. In this work, we present and demonstrate that with an *appropriate* semantic representation of the activity, and without such complex perception systems, it is sufficient to infer human activities from videos. First, we will present a method to extract the semantic rules based on three simple hand motions, i.e. *move*, *not move* and *tool use*. Additionally, the information of the object properties either *ObjectActedOn* or *ObjectInHand* are used. Such properties encapsulate the information of the current context. The above data is used to train a decision tree to obtain the semantic rules employed by a reasoning engine. This means, we extract *lower-level* information from videos and we reason about the intended human behaviors (*high-level*). The advantage of the abstract representation is that it allows to obtain more generic models out of human behaviors, even when the information is obtained from different scenarios. The results show that our system correctly segments and recognizes human behaviors with an accuracy of 85%. Another important aspect of our system is its scalability and adaptability toward new activities, which can be learned *on-demand*. Our system has been fully implemented on a humanoid robot, the iCub to experimentally validate the performance and the robustness of our system during *on-line* execution of the robot.**

I. Introduction

Humans have amazing capabilities to learn new skills by extracting and fusing new information from the environment. We can integrate and adapt the new information into our previously learned model using our cognitive capabilities, for example: perception, reasoning, prediction, learning, planning, etc. In other words, we are able to adapt toward new situations because we re-used the learned models to infer unknown activities instead of just reproducing the observed motions. Thus, to the extend that we understand, *what* we are doing. Namely, we extract the semantics of the observed behavior. Then, the ideal scenario is to transfer such capabilities to robots so that they can better learn from us.

Automatically segmenting and recognizing an activity from videos is a challenging task, mainly because the execution of a similar activity could be performed in many different manners depending on the person or the place. For example, if I prepare a pancake in my kitchen, then I may follow a predefined pattern [1]. On the other hand, if I prepare a pancake in my office's kitchen under time pressure, then I will follow another pattern even though I execute the same task. These patterns are sometimes defined by different parameters, e.g. different speeds of execution, the height of the pancake mix to pour over the stove, how much force do I need to eject in order to open a bottle, after how much time do I need to flip the dough, etc. as investigated in [2].

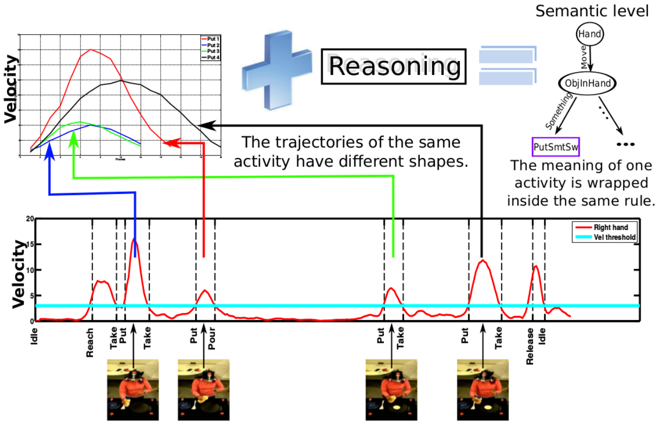

Fig. 1. This figure shows the organization of our system. Here we show an example of the analysis of real data for the *Put* activity over time.

In order to understand the observed activity, first we need to filter out the information from the input sensors to identify which factors make one activity different from the others. For example, Fig. 1 shows that we may follow different patterns to achieve similar activities. Then, what are the factors that allows us to identify the same activity, even when different patterns are observed? How can we generalize those behaviours under different situations?

One typically analyzed signal is the velocity profile to recognize human motions between *move* and *not move* [3]. However, when dealing with complex tasks such as: reach, take, put, pour, etc., then that information would no longer be sufficient because, as shown in Fig. 1, the velocity signal of the same activity could have different length, amplitude, shape, etc. over time. Another signal that could be analyzed is the distance between the hand and the object(s), or the orientation of the objects, etc. That means that by observing we can retrieve a lot of information. Then, some questions arise, whether will the correct human activity recognition depends on having the right input information? Or does it depend on having a better way of interpreting the incoming data? In this work, we will demonstrate that the information such as hand velocity and distance alone are not enough for segmenting and recognizing human activities.

[1] Faculty of Electrical Engineering, Institute for Cognitive Systems, Technical University of Munich, Germany karinne.ramirez@tum.de and gordon@tum.de

[2] Institute for Artificial Intelligence, University of Bremen, Germany beetz@cs.uni-bremen.de

In this paper, we propose a framework that combines the information from different signals via semantic reasoning to enable robots to segment and recognize human activities by understanding what it sees from videos (see Fig. 1). The contributions of this paper are: a) *on-line* segmentition of human motions from videos, b) automatic recognition of human activities using semantic representations, c) our system is adaptable and intuitive to new situations due to the re-usability of the learnt rules, d) the system is scalable because it can learn and identify new activities *on-demand*, e) our system preserves its accuracy and robustness within the "on-line" control loop of a robot. This paper is organized as follows: Section II describes the related work.Afterward, Section III introduces the methodology and results for the object recognition. Then, Section IV presents the core of our framework which is the semantic rules methodology and its results. Finally, Section V shows the robustness of our framework when implemented into the iCub.

II. Related Work

Recognizing human activities is currently an active research area in Computer Vision, where the local image representation is considered as a promising direction compared to the global representations because it can generalize to different scenarios by taking into account spatio-temporal correlation between patches [4]. However, the action analysis is focused on the movement or/and change of posture, such as walking, running, swinging, etc. [5]. Nevertheless, those approaches are used only to recognize the activities but not to segment them, i.e. the segmentation is done manually.

Another direction for action recognition has been proposed through the recognition of the object(s), human motions and the effects on the objects. Regarding the object recognition approach, the work presented by [6], shows a model that can generalize from object instances to their classes by using abstract reasoning. However, activies such as *doing laundry* and *getting dressed* are misclassified because they have the same class of object. Then, [7] introduced the concept of Object-Action Complexes (OACs), to transform objects by actions, i.e. *how object A (cup-full) changes to object B (cup-empty) through the execution of Action C (drinking)*. Recently has been used to segment and recognize an action from a library of OACs to enable a robot to reproduce the demonstrated activity [8] using a robust perception system which is executed off-line.Analogous to OACs and based on the affordance principle, [9] presented the*Semantic Event Chain*, which determines the interactions between the hand and the objects, expressed in a *rule-character* form, which also depends on a precise vision system.

Regarding the problem of recognition and understanding of human activities a few related works can be found such as the noticeable work presented by [10], which maps the continuous real world events into symbolic concepts by using an active attention control system. Another work, presented by Fern et. al. [11], introduced a logic sub-language learning specific-to-general event definitions by using manual correspondence information. Similarly, the one presented in [12], introduced a system that can *understand* actions based on their consequences, e.g. split or merge. Nevertheless, the core of this technique lies in a robust active tracking and segmentation method to detect the object changes, i.e. the consequences of the action. Later, they include a library of plans composed of primitive action descriptions [13]. However, this system is not implemented in a robot and it will fail if the plan is not known a priori. Another work based on plan recognition presented by [14] state that human behavior follows stereotypical patterns that could be expressed as preconditions and effects. However, these constraints must be specified in advance. Then, [15] shows a (partially) symbolic representation of manipulation strategies to generate robot plans based on pre- and post- conditions. Nevertheless, these frameworks are not able to either reason about the intentions of the users or extract the meaning of the actions.

In the robotics community, there has been a tendency to use the trajectory level, i.e. the Cartesian and Joint spaces, to segment and imitate human motions. For example, [3], proposed an approach to encode observed trajectories based on Hidden Markov Models (HMMs) mimesis model in order to segment and generate motions through imitation. [16] presented a Hierarchical action model constructed from observed human tracking data based on the linear-chain Conditional Random Fields (CRF) which uses pose-related features. Another technique used to classify human motions is based on the shape of the trajectory, e.g. using similarity measurements like Dynamic Time Warping [17]. These later techniques realized on the generation of trajectories depending on the location of the objects, then if a different environment is analyzed then the trajectories will be altered completely, thus, new models have to be acquired.

The architecture of our framework is inspired by [18] and contains three main modules: 1) extract the relevant aspects of the task; 2) process the perceived information to infer the goal of the demonstrator; and 3) transfer the goal to the robot to achieve the desired goal (see Fig. 1). In this paper, we demonstrate that our system performs very accurately (around 85%) even when new activities are tested; thus demonstrating that the inferred representations are not depended on the performed task. Furthermore, the robot is able to recognize *new* activities and learn the correct rule(s) *on-line*, which means that we do not need to provide it with all possible activities, which would not be possible.

III. Extraction of visual features

First, we segment the continuous video streams into meaningful classes, which is a challenging task as expressed in [4]. Then, we propose to split the complexity of the recognition in two parts. The first one will gather (perceive) information from the objects using a simple color-based technique. Whereas the second part will handle the difficult problem of interpreting the perceived information into meaningful classes using our inference module (see Section IV).

The highest level of abstraction to be segmented from videos is the hand motions, into mainly three categories:

- *move*: The hand is moving, i.e. $\dot{x} > \varepsilon$
- *not move*: The hand stop its motion, i.e. $\dot{x} \rightarrow 0$
- *tool use*: Complex motion, the hand has a tool and it is acted on a second object, i.e. $o_h(t) = knife$ and $o_a(t) = bread$

Notice, that those kind of motions can be recognized in different scenarios, but they can not define an activity by themselves. Therefore, we need to add the object information, i.e. the motions together with the object properties have more meaning than separate entities. The properties that can be recognized from the videos are:

- *ObjectActedOn (o_a)*: The hand is moving towards an object, i.e. $d(x_h, x_o) = \sqrt{\sum_{i=1}^{n}(x_{h_i} - x_{o_i})^2} \rightarrow 0$
- *ObjectInHand (o_h)*: The object is in the hand, i.e. o_h is currently manipulated, i.e. $d(x_h, x_o) \approx 0$.

The output of this module determines the current state of the system (s), which is defined as the triplet $s = \{m, o_a, o_h\}$. The definition and some examples of the motions and object properties are further explained in [19].

A. Color-based recognition methodology

To recognize the hand motions and object properties, we implemented a well-known and simple color-based algorithm. We use the OpenCV library to obtain the color features (f_v) in order to get the hand position (x_h). Then, we smooth the signal with a *low-pass* filter:

$$y_s(i) = \frac{1}{2N+1}(y(i+N) + y(i+N-1) + ... + y(i-N)) \quad (1)$$

where $y_s(i)$ is the smoothed value for the ith data point, N is the number of neighboring data points on either side of $y_s(i)$, and $2N + 1$ is the size of the moving window, which must be an odd number.

Then, we used a velocity threshold (see Fig. 3) to segment between *move* or *not move* and to recognize the *tool use* motion we need to identify the object properties, i.e. *ObjectActedOn* or *ObjectInHand*, explained in Algorithm 1

It is important to notice that the recognized object (o) can only satisfy one of the above object properties, i.e. $o_a(t) = o$ or $o_h(t) = o$ but not both at the same time t. Nevertheless, it is possible to have more than one object on the scene, for instance $o_1 = pancake$ and $o_2 = spatula$ where the object properties could be $o_a(t) = o_1$ and $o_h(t) = o_2$, then the hand motion is segmented as *tool use*.

B. Results of Color-based Recognition

We tested this methodology in two data sets: pancake and sandwich making. The first one contains recordings of one human making a pancake several times. The second data set contains a more complex activity, which is making a sandwich performed by several subjects under two time conditions, i.e. normal and fast.

In this work we use from the sandwich scenario the task of "cutting the bread" and from the pancake scenario the task of "pouring the pancake mix", as shown in the attached video. This means that each of these tasks were segmented into three motions: *move, not move* or *tool use*, as well as the object properties into *ObjectActedOn* and *ObjectInHand*.

Quantitatively, the results indicate that the human motions are correctly classified for the pancake making with 91% accuracy and for the sandwich making around 86.24% with respect to the ground-truth[1]. Regarding the recognition of the object properties, the accuracy for the pancake making is around 96.22% and for the sandwich scenario is 89.24%. The above segmentation is performed for *on-line* videos.

Algorithm 1 Object properties recognition algorithm.

```
Require: $smooth\_x[i], smooth\_y[i]$: position of the hand and objects detected.
    $threshold\_distMax$ : maximum distance between the hand and object.
    $threshold\_distMin$ : minimum distance between the hand and object.
$hand\_pos = [smooth\_x[1], smooth\_y[1]]$ {The hand is always the first object detected}
for $i = 2$ to $N$ step 1 do
    $object\_pos = [smooth\_x[i], smooth\_y[i]]$
    $distance[i-1] = getDistance(hand\_pos, object\_pos)$ {obtain the distance between the hand and the identified objects}
    $smooth\_dist[i-1] = smoothDistance(distance[i-1])$ {Apply a low pass filter to smooth the data}
end for
for $j = 1$ to $N - 1$ step 1 do
    {Find the properties of the objects on the scene}
    if $(smooth\_dist[j]) < threshold\_distMax)$ then
        $o_a = j$
        if $(smooth\_dist[j]) < threshold\_distMin$ then
            $o_h = j$
            if $(smooth\_dist[j+1] < threshold\_distMax)$ then
                $o_a = j + 1$
                $motion = TOOL\ USE$ {Tool use motion is defined if it has both properties $o_a$ and $o_h$}
            else
                $o_a = NONE$
            end if
        else
            $o_h = NONE$
        end if
    else
        $o_a = NONE$ and $o_h = NONE$
    end if
end for
return $motion,\ o_a,\ o_h$
```

IV. SEMANTIC REASONING

Semantics is defined as the *study of the meaning*. Therefore, in this paper, *the semantics of human behavior* refers to find a meaningful relationship between human motions and object properties in order to understand the activity performed by the human. In other words, *the semantics of human behavior* is used to interpret visual input to understand human activities. This has the advantage to transfer the extracted *meaning* into new scenarios.

This module represents the core and most important part of our work. Because this module will interpret the visual data obtained from the perception module and process that information to infer the human intentions. This means that it receives as input information the hand motion segmentation (m) and the object properties (o_a or o_h). In other words, it will be responsible of identifying and extracting the meaning of human motions by generating semantic rules that define and explain these human motions, i.e. it will infer the *high-level* human activities, such as: *reach, take, pour, cut,* etc.

[1] The ground-truth data is obtained by manually segmenting the videos into hand motions, object properties and human activities.

A. Semantic rules methodology

A decision tree classifier is used to learn the mapping between the *low-level* motions and the *high-level* activities through its object properties. In order to learn the decision tree we require a set of training samples S. Each sample describes a specific state of the system $s \in S$. The set S is represented by its attributes A and its target training concept value $c(s)$ for s. In other words, the training example S is an ordered pair of the form $\langle s, c(s) \rangle$ called *state-value pairs*. In this work the *training samples* S are described by the following attributes:

1) *Hand_motion* (Move, Not_move, Tool_use)
2) *ObjectActedOn* (Something, None)
3) *ObjectInHand* (Something, None)

and the *target concept value*:

- Class c : *ActivityRecognition* : $S \rightarrow$ {Reach, Take, Release, Put_Something_Somewhere, Idle, Granular[2].}

Some examples of the *state-value pair* ($\langle s, c(s) \rangle$) are:

$$\langle \{\ Move\ ,\ \ Something\ ,\ None\ \},\quad Reach\quad \rangle$$
$$\langle \{Not\_Move,\ None,\ Something\},\quad Take\quad \rangle$$

In order to learn the target function c from a set of training samples S, we use the C4.5 algorithm [20] to compute the decision tree. with the information gain measure:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{S} Entropy(S_v) \quad (2)$$

where $Values(A)$ is the set of all possible values of the attribute A, and $S_v = s \in S | A(s) = v$.

B. Semantic reasoning results

The Weka data mining software is used to generate the decision tree and the sandwich-making scenario was chosen as the training data set, because it has a high complexity due to the several sub-activities that it contains. During the training stage, we split the learning procedure in two steps. The first step will generate a tree that can determine the human *basic* activities in a general manner. The second one will extend the tree to recognize the granular activities based on the current context.

For the first step, we use the information of the ground-truth data of a subject during the normal condition while making a sandwich. We split the data as follows: 60% was used for training and 40% for testing. Then, we obtain the tree $T_{sandwich}$ shown in the top part of Fig. 2 where the following human *basic* activities can be inferred: *idle, take, release, reach, put* something somewhere and *granular*. This learning process will capture the general information between the objects, motions and activities. It is important to notice that the first attribute that has to be correctly segmented is the hand motion, e.g. if the hand is *not moving* we could predict that the activity is either *take* or *idle*, which will be defined by the object property *ObjectInHand*. This means that from the obtained tree we can determine six hypotheses ($H_{sandwich}$) which represent the semantic rules that describes the basic human activities. For example:

$$if\ Hand(Move) \& ObjectInHand(None) \& \quad (3)$$
$$ObjectActedOn(Something) \rightarrow Activity(\textbf{Reach})$$
$$if\ Hand(Tool\_use) \rightarrow Activity(\textbf{GranularActivity}) \quad (4)$$

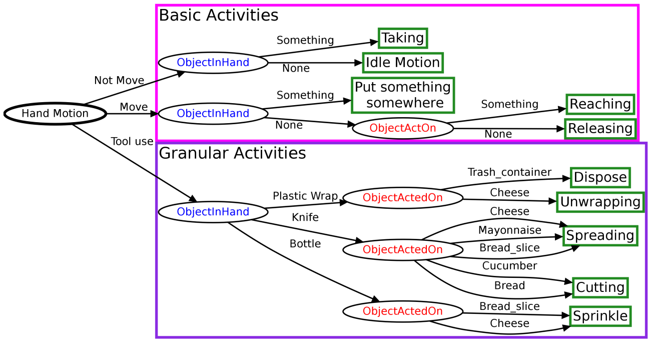

Fig. 2. This figure shows on the top part (magenta box) the tree obtained from the sandwich making scenario ($T_{sandwich}$). On the bottom (purple box) is shown the extension of the tree to infer *granular* activities.

From the sandwich making data set, activities such as: cut, sprinkle, spread, etc. are expected. However, those activities are not considered as *basic* human activities, rather as *granular* activities. Such complex activities are replaced in the input data set (s *or* n) with the label of *GranularActivity* and they are inferred with the rule shown in eq. (4). Then, to correctly infer those complex activities more attributes have to be considered, e.g. we can take into account the type of object being manipulated, for example for cut and spread, they both use the *knife* as a tool but they represent different activities, defined by the object they are acted on (o_a), either the bread or the mayonnaise, respectively. Therefore, a second stage is needed in order to extend our tree T and be able to infer those *granular* activities.

For the second step, we use as input the activities clustered as *GranularActivity* from the previous step and we learn a new tree, which represents the extension of our previous tree. The final tree can be observed in Fig. 2, where the top part (magenta box) represents the general and most abstract level of rules to determine different *basic* activities and the bottom part (purple box) presents the extension of the tree, given the current information of the objects. This means that, in order to identify which *granular* activity is being executed, we need to know which objects are being identified. Notice, that the taxonomy of the tree is obtained which will allow us to add new rules when a new activity is detected.

Then, the next step is to test the accuracy of the obtained tree $T_{sandwich}$. In order to do that, we use the remaining 40% of the sandwich data set to test the accuracy of the obtained rules. In other words, given the input attributes $n_{sandwich\_test} = \{Move,\ Something,\ None\}$ we will determine $c(n_{sandwich\_test})$. Then, the *state-value* pairs from the test data set $n_{sandwich\_test}$ will be of the form $\langle n_{sandwich\_test}(t), ? \rangle$, where t represents the time $(frames)$.

[2] Granular activities define classes such as flip, pour, cut, etc. These activities are difficult to generalize because they depend on the context.

Afterward, the target value is determined for each state of the system $c(n_{sandwich\_test}(t))$. Finally, the obtained results show that $c(n_{sandwich\_test}(t))$ was correctly classified 92.57% of the instances using as input information manually labeled data, i.e., during the *off-line* recognition. A similar tree is obtained if the training set is the pancake-making [21].

1) Action recognition using Color-base: The next step is to use as input the data obtained from the automatic segmentation of human motions and object properties, in order to test the *on-line* recognition (see Section III-B). First, we applied the learned rules to a known scenario using the same task as the trained one, i.e. sandwich making. In order to test the semantic rules we use a different subject than the one used for the training and two conditions were tested: normal and fast. The results show that the accuracy of recognition is about 81.74% (Normal condition= 79.18% and Fast condition=83.43%). The errors in the activity recognition are because of the misclassified objects from the perception module, specially for the sandwich scenario, when the object *knife* disappears between the hand and the bread (see Fig. 3).

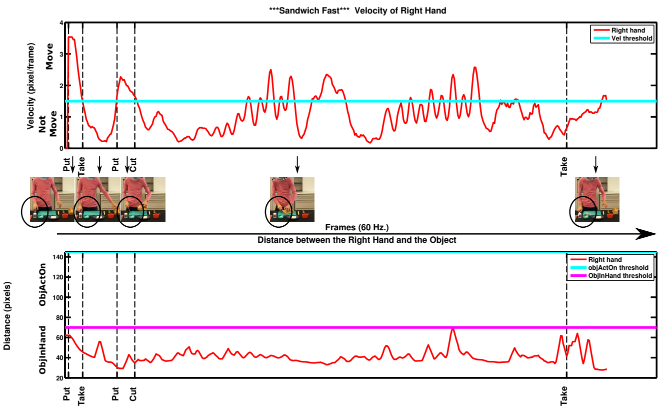

Fig. 3. Depicts the output signals of the hand and object tracker of the sandwich scenario when the subjects is in a speed condition. The vertical lines indicate the automatic segmentation and recognition of the human activities for the right hand.

Then, we tested the semantic rules into a new scenario, in which the activity *pour* has not yet been learned. Nevertheless, the system is able to identify that a new activity has been detected and asks the user to name the *unknown* activity as shown in the attached video. Then, the new activity has been learned and the system can correctly infer it. The results indicate that the accuracy of recognition is around 88.27%.

The important contribution of these results is the definition of rules that make the inference of human activities in different scenarios possible, with an overall accuracy of 85%, considering known and unknown activities. The above is possible even with a very simple hand and object recognition to segment the motions and object properties automatically.

V. Experimental Integration and Validation

Finally, we validate our *on-line* segmentation and recognition in a robotic system, in this case the iCub. The iCub is a 53 degrees of freedom humanoid robot [22] and its strong humanoid design provides an appropriate testing platform.

One important factor to consider is the transition from *off-line* learning to *on-line* learning. The perception and semantic modules can easily be implemented for *off-line* systems as we have shown in [19]. However, for *on-line* systems we have to consider the possibility of learning new activities *on-demand* as we proposed in the previous sections. Additionally, the perception and semantic systems need to be as fast and accurate as possible. In other words, the communication between the perception and inference modules have to be instantaneous because these modules has to be implemented inside the control loop of the robot (see Fig. 4).

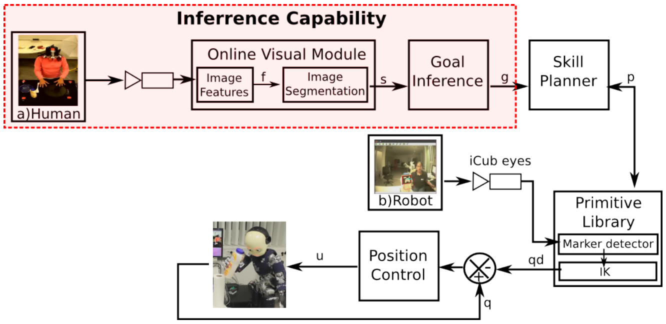

Fig. 4. Integration of the system into the control block of the iCub. This process includes information from external views obtained from videos (a) and environment information obtain from the iCub's cameras (b).

The red highlighted square of Fig. 4 depicts the inclusion of the perception and semantic modules into the robot. The flow of the control loop of the robot is as follows: a) First, the video streams the desired activity and the *low-level* motions and object properties are automatically segmented. b) Immediately the semantic system will retrieve the inferred activity. c) Finally, the inferred activity will trigger the plan and the motion primitives that the robot needs to execute in order to achieve a similar goal as the one observed. Noticeable, all the modules receive inputs and produce the desired outcome *on-line*. In other words, first the robot watches the video, then it understand the activity and finally it produce the corresponding motion, as shown in Fig. 5.

Regarding the skill execution by the robot, the system works as follows: from the inferred activity, there is a module that will select which execution plan will be performed. Then, the plan will indicate the motion primitives that the robot needs to execute in order to achieve a similar goal as the human. For example, if the inferred activity is *reaching*, then a position-based visual servoing module is executed. This module will extract 2D visual (image) features from a stereo vision system with AR markers. We use the ArUco library which is based on OpenCV to detect markers. 3D position and rotation with respect to the camera frame (iCub right eye) are obtained from the image features using the camera intrinsic parameters. Once, the marker is detected, the next primitive is to move the right arm of the robot toward the desired Cartesian position. This is achieved using the inverse kinematics. Beside pick and place activities such

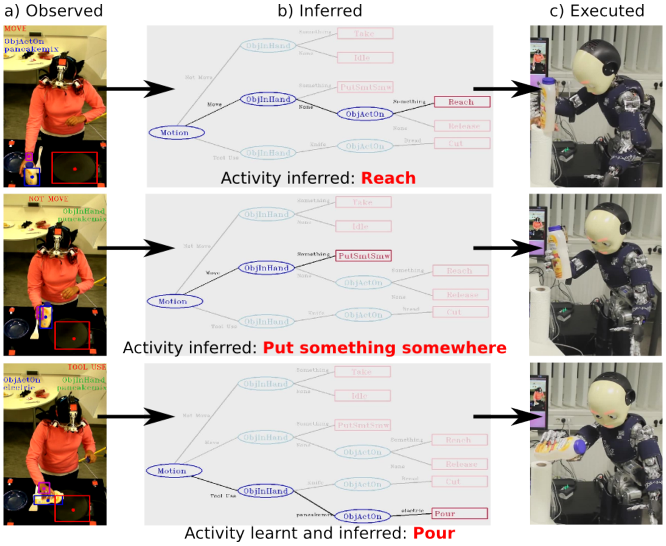

Fig. 5. First the robot observes the motions of the human from a video, then it infers or learns the human activity and finally the iCub execute a similar activity.

as reach, take, put something somewhere and release, our system can handle more specific activities such as pouring, which are shown in the attached video.

The modular architecture of our framework allows to replace any module to acquire more complex behaviors, e.g. the vision module can be replaced for a more advance detection system or the control approach can be substitute by a more robust and adaptive control law, e.g [23].

VI. Conclusions

Correctly identifying human activities is a challenging task in the robotics community, and its solution is very important because it is the first step toward a more natural human-robot interaction. In this paper we present a methodology to extract the meaning of human activities by combining the information of the hand motion and two object properties. Our proposed framework has a classification accuracy for *on-line* segmentation and recognition of human activities of 85% even when a very simple perception system is used for real, challenging and complex task.

Additionally, our framework is possible to be integrated and executed *on-line* within the control-loop of a robotic system. Further advantages of our system are its scalability, adaptability and intuitiveness which allow a more natural communication with artificial system such as robots.

Acknowledgments

The work leading to these results has received funding from the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 609206 and it was supported (in part) by the DFG cluster of excellence *Cognition for Technical systems CoTeSys*.

References

[1] D. Wolpert and Z. Ghahramani, "Computational principles of movement neuroscience," *Nature Neuroscience Supplement*, vol. 3, pp. 1212–1217, 2000.

[2] L. Kunze, M. E. Dolha, and M. Beetz, "Logic programming with simulation-based temporal projection for everyday robot object manipulation." in *IROS*. IEEE, 2011, pp. 3172–3178.

[3] W. Takano and Y. Nakamura, "Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation," in *Humanoids*. IEEE, 2006, pp. 425–431.

[4] R. Poppe, "A survey on vision-based human action recognition." *Image Vision Comput.*, vol. 28, no. 6, pp. 976–990, 2010.

[5] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis." in *CVPR*. IEEE, 2011, pp. 3361–3368.

[6] D. J. Patterson, D. Fox, H. A. Kautz, and M. Philipose, "Fine-Grained Activity Recognition by Aggregating Abstract Object Usage." in *ISWC*. IEEE Computer Society, 2005, pp. 44–51.

[7] F. Wörgötter, A. Agostini, N. Krüger, N. Shylo, and B. Porr, "Cognitive agents - a procedural perspective relying on the predictability of Object-Action-Complexes (OACs)." *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 420–432, 2009.

[8] M. Wächter, S. Schulz, T. Asfour, E. Aksoy, F. Wörgötter, and R. Dillmann, "Action Sequence Reproduction based on Automatic Segmentation and Object-Action Complexes," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2013.

[9] E. E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, and F. Wörgötter, "Learning the semantics of object-action relations by observation." *I. J. Robotic Res.*, vol. 30, no. 10, pp. 1229–1249, 2011.

[10] 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.

[11] A. Fern, J. M. Siskind, and R. Givan, "Learning Temporal, Relational, Force-Dynamic Event Definitions from Video." in *AAAI/IAAI*, R. Dechter and R. S. Sutton, Eds. AAAI Press / The MIT Press, 2002, pp. 159–166.

[12] Y. Yang, C. Fermüller, and Y. Aloimonos, "Detection of Manipulation Action Consequences (MAC)." in *CVPR*. IEEE, 2013, pp. 2563–2570.

[13] A. Guha, Y. Yang, C. Fermuuller, and Y. Aloimonos, "Minimalist plans for interpreting manipulation actions," in *IROS*, 2013, pp. 5908–5914.

[14] H. A. Kautz, H. A. Kautz, R. N. Pelavin, J. D. Tenenberg, and M. Kaufmann, "A formal theory of plan recognition and its implementation," in *Reasoning about Plans*. Morgan Kaufmann, 1991, pp. 69–125.

[15] R. Jäkel, S. R. Schmidt-Rohr, M. Lösch, and R. Dillmann, "Representation and constrained planning of manipulation strategies in the context of Programming by Demonstration." in *ICRA*. IEEE, 2010, pp. 162–169.

[16] M. Beetz, M. Tenorth, D. Jain, and J. Bandouch, "Towards Automated Models of Activities of Daily Life," *Technology and Disability*, vol. 22, 2010.

[17] S. Albrecht, K. Ramirez-Amaro, F. Ruiz-Ugalde, D. Weikersdorfer, M. Leibold, M. Ulbrich, and M. Beetz, "Imitating human reaching motions using physically inspired optimization principles." in *Humanoids*. IEEE, 2011, pp. 602–607.

[18] M. Carpenter and J. Call, "The question of what to imitate: inferring goals and intentions from demonstrations," in *Imitation and Social Learning in Robots, Humans and Animals.*, K. Dautenhahn and C. L. Nehaniv, Eds. MIT Press, 2007.

[19] K. Ramirez-Amaro, E.-S. Kim, J. Kim, B.-T. Zhang, M. Beetz, and G. Cheng, "Enhancing Human Action Recognition through Spatio-temporal Feature Learning and Semantic Rules," in *Humanoid Robots, 2013, 13th IEEE-RAS International Conference*, October 2013.

[20] R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[21] K. Ramirez-Amaro, M. Beetz, and G. Cheng, "Extracting Semantic Rules from Human Observations." in *ICRA workshop: Semantics, Identification and Control of Robot-Human-Environment Interaction.*, May 2013.

[22] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The iCub humanoid robot: an open platform for research in embodied cognition," in *PerMIS*, 2008, pp. 19–21.

[23] E. C. Dean-Leon, V. Parra-Vega, and A. Espinosa-Romero, "Global Uncalibrated Visual Servoing for Constrained Robots Working on an Uncalibrated Environments." in *IROS*. IEEE, 2006, pp. 3809–3816.