To appear in *Advanced Robotics*
Vol. 00, No. 00, March 2015, 1–23

FULL PAPER

Understanding the Intention of Human Activities through Semantic Perception: Observation, Understanding and Execution on a Humanoid Robot

Karinne Ramirez-Amaro[a]*, Michael Beetz[b] and Gordon Cheng[a]

[a]*Faculty of Electrical Engineering, Institute for Cognitive Systems, Technical University of Munich, Germany*; [b]*Institute for Artificial Intelligence, University of Bremen, Germany*

(*Received 00 Month 201X; accepted 00 Month 201X*)

In this work, we present and demonstrate that with an *appropriate* semantic representation and even with a very naive perception system, it is sufficient to infer human activities from observations. First, we present a method to extract the semantic rules of human everyday activities. Namely, we extract *low-level* information from the sensor data and then we infer the *high-level* by reasoning about the intended human behaviors. The advantage of this abstract representation is that it allows us to obtain more generic models from human behaviors, even when the information is obtained from different scenarios. 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 within the control loop of the robot. The results show that the robot is able to make a decision in 0.12 *seconds* about the inferred human behaviors with a recognition accuracy of 85%.

Keywords: automatic segmentation; semantic reasoning; human activity recognition; meaningful robot learning.

1. Introduction

One of the main purposes of humanoid robots is to improve the quality of life of elderly and/or disabled people by helping them in their everyday activities. Therefore, such robotic systems should be flexible and adaptable to new situations. This means that they need to be equipped with cognitive capabilities such as perception, learning, reasoning, planning, etc [1]. These capabilities could enable robots to segment, recognize and understand *what the demonstrator is doing* by observation [2]. Thus, to the extend that the robot can understand the observed behavior.

Transferring skills to humanoid robots from observing human activities is well considered to be one of the most effective ways to increase the capabilities of such systems [3, 4]. With the recent advancements of sensory technologies (such as Kinect), perceiving reliably human activities have become tenable [5]. If robots are expected to learn or interact with humans in a meaningful manner, the next foreseeable challenge for the robotic research in this area is toward the semantic understanding of human activities - enabling them to extract and determine *higher level* understanding. The ability to automatically recognize human behaviors and react to them by generating the next probable motion or action according to human expectations will enrich humanoid robots substantially.

*Corresponding author. Email: karinne.ramirez@tum.de

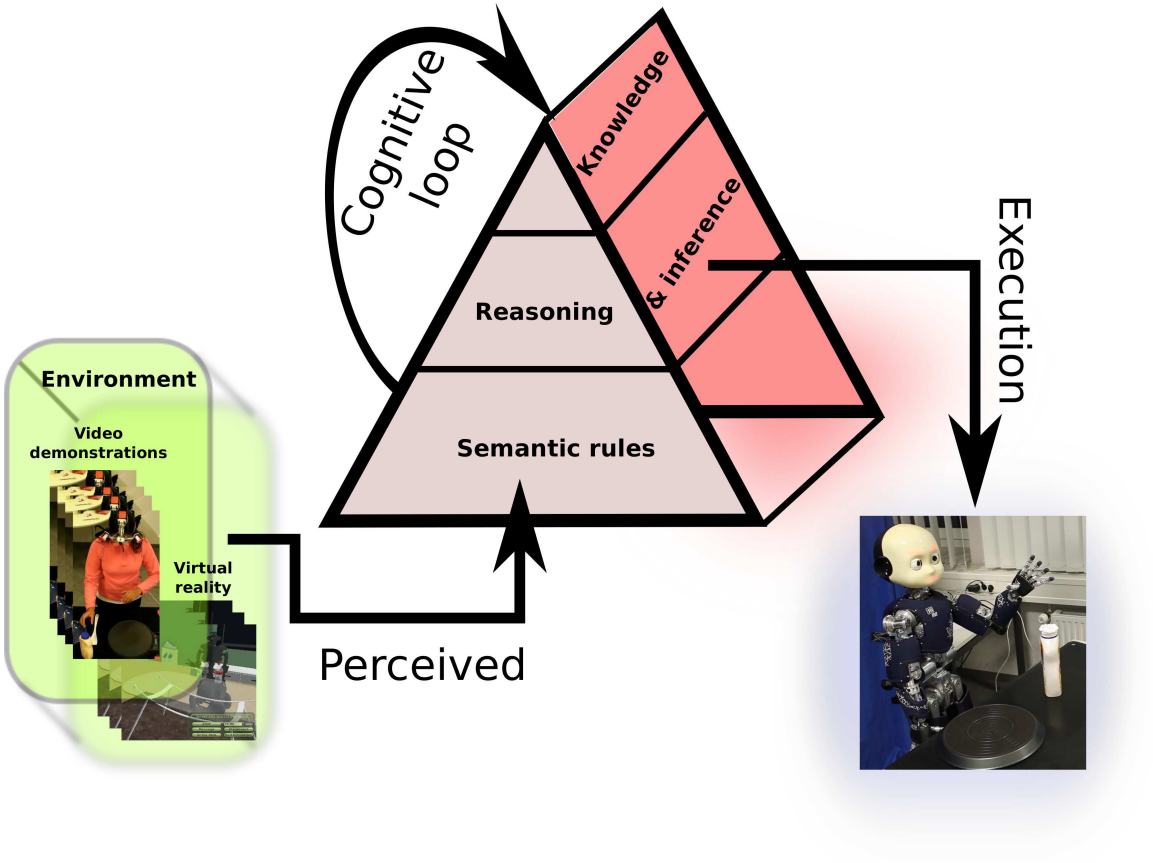


Figure 1. Conceptual diagram of the modules implemented in our system for the understanding of human activities. First, we perceive the environment information from different sources. Then, we extract the *important* features to infer the observed activity, this implies that the cognitive loop contains the capabilities of reasoning, semantics, knowledge, etc. Finally, the robot will execute a motion primitive to achieve a similar goal as the one inferred.

In this work, we propose a framework that combines several observable inputs together with suitable reasoning tools to properly interpret, learn and understand human behaviors from demonstrations (see Fig. 1). This framework was first introduced in [6] and its robustness was tested with manually labeled data. Then, in [7] we improved our framework with the inclusion of three videos, showing the same demonstration from different views and using an unsupervised state-of-the-art learning algorithm based on Independent Subspace Analysis (ISA) [8] to extract spatio-temporal features from *off-line* videos. Recently, in [9] we introduced an *on-line* segmentation and activity recognition of one observed hand from continuous video stream implemented in a humanoid robot using our semantic representation framework. The next subsection specifies the new improvements of our system presented in this paper.

1.1 *Contributions of this paper*

Our framework can be utilized for the difficult and challenging problem of tasks and skills transfer for humanoid robots. We propose a method, that enables robots to obtain and determine *higher-level* understanding of a demonstrator's behavior via semantic reasoning. This paper presents the enhancement of our framework by including new features. For example, we introduce new situations with additional and different action types, such as: *pouring the pancake mix*, *flipping the dough* and *setting the table*. Furthermore, we present the extension of our framework to allow *on-line* recognition of parallel activities executed by both hands with a very *naive* perception system. Additionally, we experimentally validate our semantic-based framework in the control loop of a robotic platform.

Hence, demonstrating the robustness of our semantic-based framework under different condi-

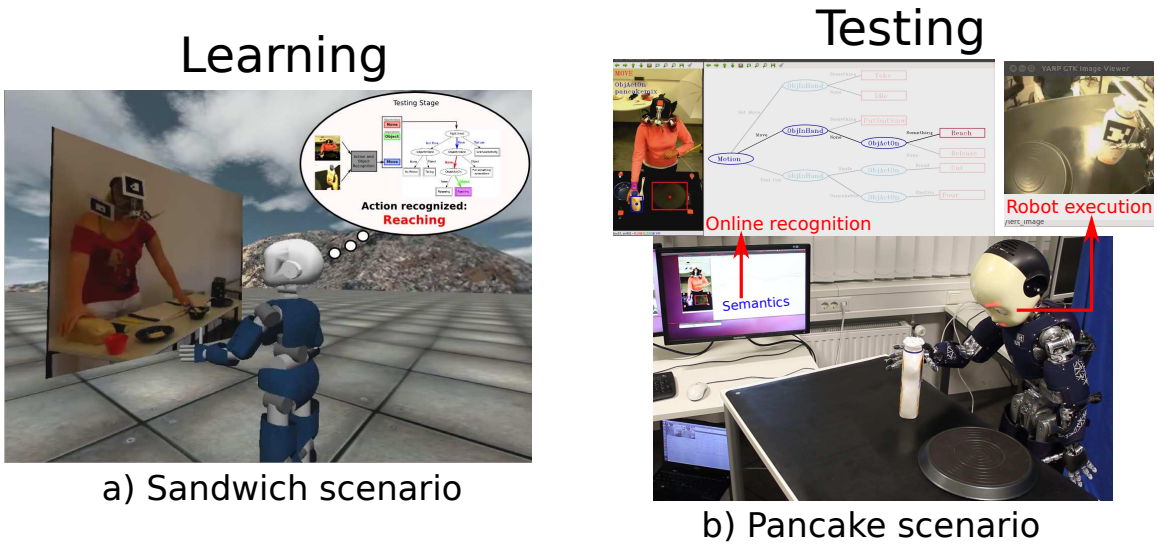


Figure 2. This figure shows the transference of knowledge between different scenarios. Two cases are depicted: a) shows the learning stage for the automatic segmentation and recognition of human activities; and b) shows that even when a different scenario is observed the *semantics* of the observed activity remains the same as a).

tions and constraints. In summary the main contributions of this work are (see Fig. 2):

- Proposed and realized a multilevel framework to automatically segment and recognize human behaviors from observations. This is achieved by extracting abstract representations of the observed task using semantic reasoning tools.
- The presented system is robust, adaptable, scalable and intuitive to new situations due to the re-usability of the learned rules.
- We propose a flexible imitation system that preserves its accuracy and robustness in the *on-line* control loop of a robot.

This paper is organized as follows: Section 2 presents the related work. Then, Section 3 explains the method to segment the visual information. Afterward, Section 4 presents the implementation of the semantic rules. Then, Section 5 explains the algorithms to integrate the modules into the iCub robot. Finally, Section 6 presents the conclusions and final remarks of this work.

2. Related work

Automatically segmenting, recognizing and understanding human activities from observations, has interested the researchers of different disciplines such as: Computer Vision [8, 10], Artificial Intelligence [11–13], Cognitive Science [14–16], Robotics [17–19], to name a few. Each of them focuses on solving a subset of the complex problem of interpreting human activities. For example, the Computer Vision community is focused on solving the problem of action recognition by identifying the important features from the images [8] or using spatio-temporal correlation [10]. Typically the action analysis is focused on recognizing the movement or/and change of posture of humans, for example using the KTH benchmark data set [20]. Another work used to recognize the human activities from observed human tracking data was presented by Beetz et al. [12] where a Hierarchical action model based on the linear-chain Conditional Random Fields (CRF) was used or when a similarity and optimization methods were used [21].

On the other hand, the Robotics community mainly investigates the problem of transferring the human behaviors into robots mainly using techniques based on the trajectory level, in order to learn and transfer human motions into robots [3], which is a very challenging task due to the embodiment problem [22]. Most of the techniques used in this community require the information of the joint and/or the Cartesian position of the human and/or the robot, as well as several trials of the same task to learn the correct model [23], for example by learning *relevant* features from

the task [24, 25], using a library of *Dynamic Motion Primitives* (DMPs) [26–28], learning a Hidden Markov Model (HMM) mimesis model [18, 29], among others.

However, recent studies focused on determining the levels of abstraction to extract meaningful information from the observed task. For example, hierarchical approaches are capable to recognize high-level activities with more complex temporal structures [11]. Such approaches are suitable for a semantic-level analysis between humans and/or objects. Extracting symbolic descriptions from observations have been proposed as a bottom-up approach to obtain the motion sequences [30]. However, this latter method is limited since it does not consider the continuity of the human sequences. One pioneer work of high-level representations was introduced by [31], where the authors suggested to map the continuous real world events into symbolic concepts using an active attention control system. Later, Ogawara et al. [32] presented a framework that integrates multiple observations based on attention points. They proposed a two-step system which observes and extracts the attention points to examining the sequence of the human motions. Then, it was proposed to use a (partially) symbolic representation of manipulation strategies to generate robot plans based on pre- and post- conditions [33], or using a logic sublanguage to learn specific-to-general event definitions with manual correspondence information [34].

Another interesting definition of semantic representations is given by [35]. The authors suggested that the semantics of human activities requires higher level representations and reasoning methods. They discussed the following approaches: *Graphical Models* (Belief Networks [36], Petri Nets [37], etc), *Syntactic Approaches* (Grammars [38], Stochastic Grammars [39], etc), *Knowledge* and *Logic Approaches* (Logic-based approaches [40], Ontologies [41], etc.). Therefore, the semantic definition of the activities depends on the used approach. For example, Graphical Models such as the one presented by [42], where a graphical model is used to learn functional object-categories. The obtained graphs encode and represent interaction between objects using spatio-temporal patterns. The taxonomy of the learned graph represents the semantics of the studied object categories mapped to a set of spatial primitives relationships, e.g. two objects are Disconnected, connected trough the Surroundings (S) or Touching (T). However, in order to obtain the *activity graph* all the episodes need to be observed. Regarding Syntactic Approaches, for instance Context-Free Grammars (CFGs) and Stochastic Context-Free Grammars (SCFGs) have been used by previous researchers to recognize high-level activities [11]. These grammars are typically used as a formal syntax for the representation of human activities. This means that these grammars directly describe the semantics of the activities.

Recently the work introduced by [43], where a system that can *understand* actions based on their consequences is proposed, e.g. split or merge using a robust active tracking and segmentation method, which can be improved by including a library of plans composed of primitive action descriptions presented by [44]. Both systems used the concept of Object-Action Complexes (OACs) [15], which investigates the transformation of objects by actions. i.e. how object A (cup-full) changes to object B (cup-empty) through the execution of Action C (drinking)[1]. This approach has been recently used to segment and recognize an action from a library of OACs using the preconditions and effects of each sub-action which enables a robot to reproduce the demonstrated activity [45]. However, this system requires a robust perception system to correctly identify the attribute of the objects, therefore it is executed off-line.

Analogous to OACs and based on the *Affordance Principle*, Aksoy et al. [14] presented the approach called *Semantic Event Chain* (SEC), which determines the interactions between hand and objects, expressed in a *rule-character* form. These interactions are based on the changes in the visual space represented in a dynamic graph where the nodes are the center of the image segments and the edges define whether or not two segments *touch* each other. Then, the spatial relationships between the graphs are stored in a transition matrix which represents the *Semantic Event Chain*. One drawback of this technique is that it highly depends on the time and sequence

[1]This action is defined by the current attribute of object A.

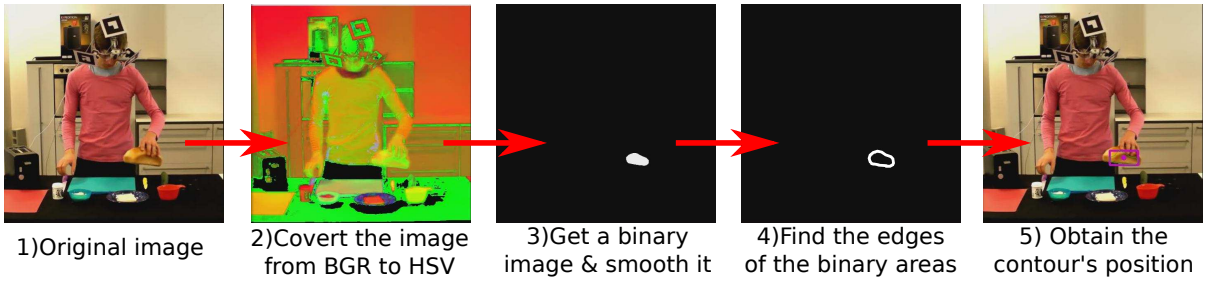


Figure 3. Color-Based pipeline to detect and track objects from an image. The outcome of this procedure is the position (x_o) of the object.

of the events, and on the perception system to define the object interactions. In other words, if the computer vision system fails, then this approach will be greatly affected, as indicated by the authors. Another approach based on the *affordances* of objects has been introduced [46], where the authors categorize the manipulated objects and human actions in context of each other, where only one hand action is considered. They have a semantic level to represent action-object dependencies (drink-cup, drink-glass, etc.) modeled using CRF method. However, this approach requires that the training data-set has been fully and correctly manually labeled, which indicates that new unlearned behaviors can not be identified.

3. Extraction of *low-level* visual features

In this work we present a new approach to successfully recognize human activities from videos using semantic representations. This means that we propose to split the complexity of the recognition problem in two parts: 1)the system recognizes the *low-level* motions (m) such as *move, not move* or *tool use* together with two object properties, e.g. *ObjectActOn*(o_a) and *ObjectInHand*(o_h); and 2) the system reasons about more specific activities (*reach, take, cut,* etc.), i.e. using the identified motions and the objects of interest from step 1.

3.1 *Pipe-line of the object recognition to obtain the low-level motions.*

In order to recognize the hand motions and object properties, we implement a well-known and simple Color-Based algorithm since this method could be applied for *on-line* object recognition for the final integration into the robot. We use the OpenCV (Open Source Computer Vision) library [47] to obtain the color visual features (f_v) in order to get the hand position (x_h) to compute its velocity ($\dot{x}_h$).

The steps to detect and track the desired object(s) are as follows: First, we convert the color space of the original image from BGR to HSV, since it is more suitable for color based image segmentation. Then, we obtain a binary image using the function *cvInRange()*, which uses the upper and lower boundary array for thresholding the image. The boundaries are obtained *off-line* and they represent the maximum and minimum limits of the HUE, SATURATION and VALUE of the object to be detected, i.e. the HSV_{min}^{max} color space, which is obtained heuristically in this work. As a result, the obtained image contains the recognized area(s) of interest represented as white isolated objects. After that, we smooth the binary image using the function *cvSmooth()* with the method *CV_MEDIAN*. Then, we use the function *cvCanny()* to find the edges of the smoothed image, followed by the function *cvFindContours()* to obtain the area enclosed by the recognized contour, where the position of the identified object x_o is retrieved. The above process is depicted in Fig. 3.

Then, we smooth the obtained position of the object x_o with a *low-pass* filter:

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

Table 1. Definition of low-level hand motions and object properties

	Name	Meaning	Formula	Example
Hand Motions	Move	The hand is moving	$\dot{x} > \varepsilon$	Moving from position A to position B
	Not move	The hand is stationary	$\dot{x} \to 0$	Holding a bread
	Tool Use	Complex motion, the hand has a tool and it is acted on a second object	$o_h(t) = knife$ and $o_a(t) = bread$	Cutting the bread where the objects are knife and bread
Obj. Prop.	Object acted on (o_a)	The hand is moving towards an object	$d(x_h, x_o) = \sqrt{\sum_{i=1}^{n}(x_h - x_{o_i})^2} \to 0$	Reaching for the bread, where $o_a(t) = bread$
	Object in hand (o_h)	The object is in the hand, i.e. o_h is currently manipulated	$d(x_h, x_o) \approx 0$	Hold/take the bread, where $o_h(t) = bread$

where $x_s(i)$ is the smoothed value for the ith data point, N is the number of neighboring data points on either side of $x_s(i)$, and $2N + 1$ is the size of the moving window, which must be an odd number. Previous literature [29] proposed to segment human motions into short sequences of motions mostly using the information of the velocity of the analyzed limbs. The segmentation of the motions is done by setting the velocity thresholds heuristically, mostly to determine if the limbs were moving or not. In a similar manner, we have proposed a procedure to segment the human motions based on the velocity of the hand(s). The segmentation of the hand motion into *move* or *not move* is done using an heuristically determined velocity threshold (ε) as shown in Fig. 7(a), however we are working on defining these thresholds automatically using the Image-Based Learning Approach (IBLA) [48].

Then, using a similar procedure we obtain the current position of the objects (x_{o_i}, $i = number\ of\ objects$) on the environment. Afterward, we compute the distance between the hand and object(s) position, i,e. $d(x_h, x_o)$. The definition and some examples of the motions and object properties are shown in Table 1. Therefore, the output of this module represents the current state of the system (s), which is defined as the triplet $s = \{m, o_a, o_h\}$.

The recognized object (o) can only satisfy one of the two object properties, i.e. $o_a(t) = pancake$ or $o_h(t) = pancake$ but not both at the same time t. Nevertheless, it is possible to have more than one object in 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*.

3.2 *Description of the used data sets*

In order to test the robustness of the generated semantic rules in different scenarios, we use three real-world scenarios: *pancake making*, *sandwich making* and *setting the table*.

3.2.1 Pancake making

First we recorded one human making pancakes nine times. The human motions are captured by three cameras located in different positions. However, for the evaluation of our framework we only use the information from camera 2 (see Fig. 4). This represents another advantage of this work compared with our previous work were the three views were required as input to the system [7].

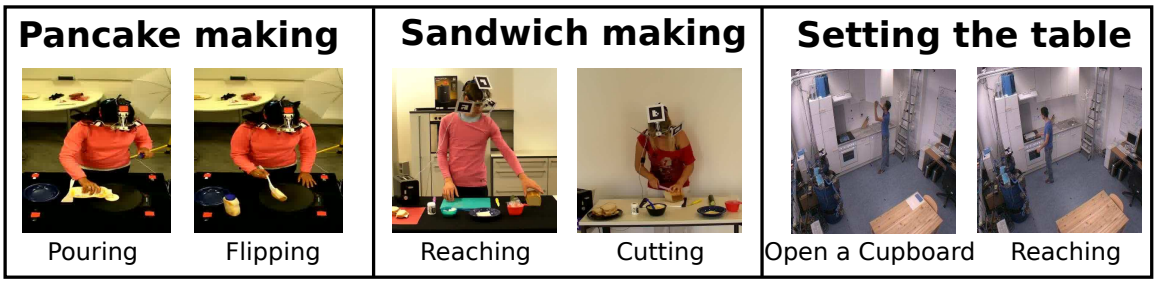


Figure 4. These figures show some snapshots of the different data sets used to test the obtained semantic rules. The first column shows the main tasks analyzed from the *pancake making* scenario. The second column shows examples of the *sandwich making* scenario, this is the data set use for training the *semantics*. The last column shows the *setting the table* data set.

3.2.2 Sandwich making

Then, we recorded a more complex activity, i.e. making a sandwich. This task is performed by eight (randomly selected) subjects, and each subject prepared approximately sixteen sandwiches, half of the sandwiches were prepared under normal time conditions and the rest under time pressure (in a hurry), see Fig. 4. The *pancake making* and *sandwich making* data sets are publicly available: http://web.ics.ei.tum.de/~karinne/DataSet/dataSet.html

3.2.3 Setting the table

Finally, we use videos from the TUM Kitchen Data Set [49], which contains observations of four subjects setting a table at least four times (see Fig. 4). The subjects were randomly selected and they performed the actions in a natural way. The human motions and object properties from this scenario were manually annotated in order to test this data set.

3.3 *Results on the automatic segmentation*

We test the Color-Based algorithm to extract human motions and object properties from two data sets: *pancake making* and *sandwich making*. The experiments were performed on a subset of all the videos. For the *pancake making* scenario, we segment the video until the *pouring the pancake mix* task and *flipping the dough* were finished. While, for the sandwich scenario, we segment the video until the *cutting the bread* task has ended. After the segmentation of the video, we execute the algorithm for two conditions: normal and fast speed.

Quantitatively the results indicate that the human motions for both hands (*move, not move, tool use*) are correctly classified for *pouring the pancake mix* with 91% accuracy, for *flipping the dough* 86.92% accuracy and for *cutting the bread* around 86.24% with respect to the ground-truth[1]. Examples of the obtained confusion matrix[2] of the human motions are shown in Table 2. Regarding the recognition of the object properties for both hands (*ObjectActedOn* and *ObjectInHand*), the accuracy for *pouring the pancake mix* is around 96.22%, for *flipping the dough* 90.65% accuracy and for *cutting the bread* is 89.24%.

Figure 5, depicts the obtained signals from the Color-Based tracking system. It is possible to observe that the obtained trajectories of both hands are very different from each other, however our system is able to segment the hand motions into three categories: *move* (blue line), *not move* (green line) and *tool use* (red line).

We can notice that even though we are using a very simple algorithm to identify and track objects from videos, the obtained accuracy is high. Furthermore, this Color-Based method is possible to apply for *on-line* object recognition, as implemented in this work. Which means that the above results were obtained for the *on-line* segmentation of videos. Nevertheless, one of the

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

Table 2. Confusion matrix of human motions expressed in % for *pouring the pancake mix* and *cutting the bread*

		Classified as											
		Right Hand						Left Hand					
		a) *pouring the pancake mix*			b) *cutting the bread*			a) *pouring the pancake mix*			b) *cutting the bread*		
		Not Move	Move	Tool Use	Not Move	Move	Tool Use	Not Move	Move	Tool Use	Not Move	Move	Tool Use
Actual Class	Not Move	**80.5**	19.46	0	**80.46**	10.59	8.94	**96.42**	3.57	0	**95.73**	4.26	0
	Move	0	**100**	0	37.76	**52.44**	9.79	0	**100**	0	19	**81**	0
	Tool Use	1.02	0	**98.97**	18.29	1.01	**80.68**	6.06	19.69	**74.24**	0	0	0

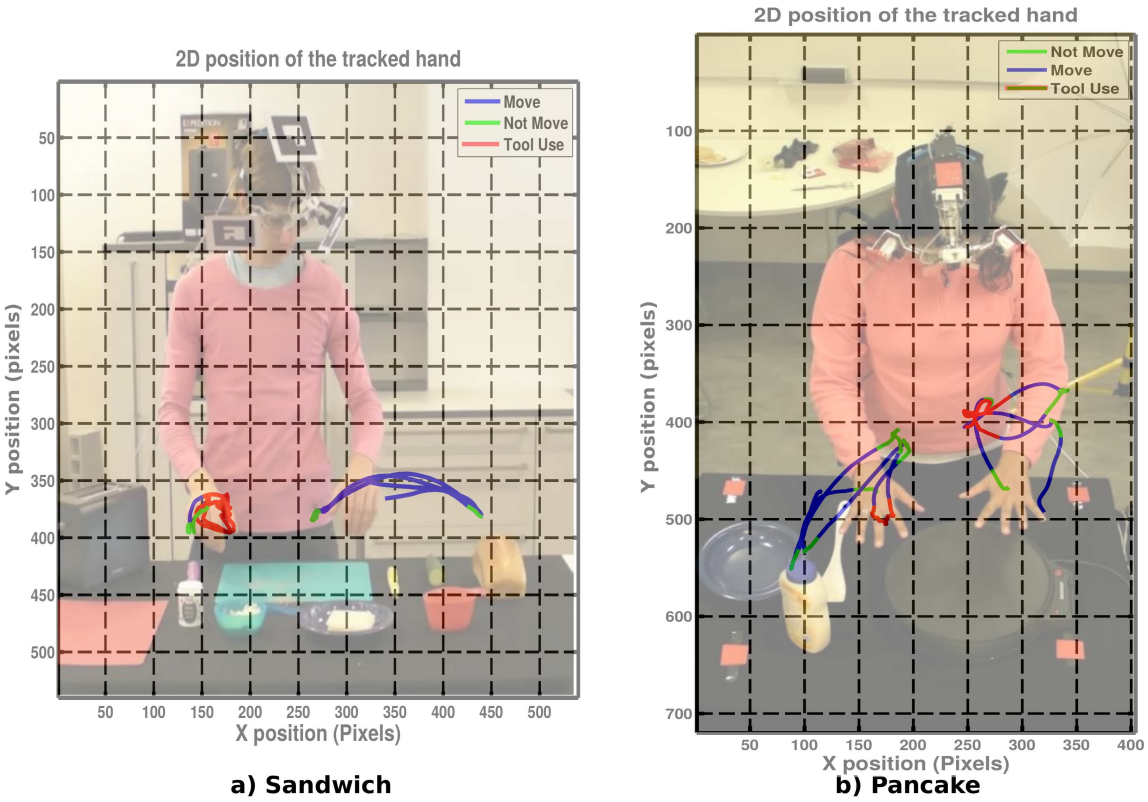


Figure 5. Results from the obtained trajectories from the Color-Based technique and the automatic segmentation in three classes: *move* (blue line), *not move* (green line) and *tool use* (red line).

limitations of the Color-Based method is that the object(s) and the background should have a significant color difference in order to successfully segment them and each object needs to have different colors. Noticeable, the segmented hand motions would not be enough for recognizing human activities such as *reaching, taking, cutting,* etc. Therefore, we need to implement the semantic reasoning engine which is described in the next section.

4. Semantic representations to infer *high-level* human activities

The goal of this section is to find the right mechanisms to interpret the human activities using semantic representations. The semantics of human behavior in this work refer 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 the visual inputs in order to understand the human activities. Our approach consists of two steps:

(1) Generate a tree that can determine the human *basic* activities in a general form, i.e. *reach, take, put, release, idle and granular* (see Section 4.1).

(2) Extend the obtained tree to recognize more *complex* activities. We call this kind of activities *granular* activities, for instance *cut, pour, spread, flip*, etc. The major difference between this kind of behaviors is the context as it is explained in subsection 4.2.

4.1 *Learning basic human activities*

For the first step, we learn a decision tree based on the C4.5 algorithm [50] from a set of training samples D. Each sample describes a specific state of the system $s \in S$. The set of instances S is represented by its attributes A and its target training concept value $c(s)$ for s. The training example D is an ordered pair of the form $\langle s, c(s)\rangle$ called *state-value pairs*. Similar to our previous work [9], the *training samples* D are described by the following attributes: $\langle\{\ HandMotion,\ ObjectActedOn,\ ObjectInHand\},\ BasicActivity\rangle$. For example,

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

The central core of the C4.5 algorithm is to select the most useful attribute to classify as many samples as possible using the information gain measure:

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

where $Values(A)$ is the set of all possible values of the attribute A, and $S_v = s \in S|A(s) = v$ as a collection of samples for S, and the entropy is defined as:

$$Entropy(S) = \sum_{i=1}^{c} -p_i log_2 p_i \tag{3}$$

where p_i is the probability of S to belong to class i. One of the advantages of using decision trees is its possibility to be represented as sets of *if-then* rules to improve human readability.

4.2 *Learning granular human activities*

In order to infer granular activities such as: *cut, pour*, etc., more attributes have to be considered. For instance, to differentiate between the activities *cut* and *spread*, which they both use the tool *knife*, these two activities acted on two different objects (o_a), either the bread or the mayonnaise, respectively. Therefore, a second stage is needed in order to extend our obtained tree T and be able to infer those *granular* activities, i.e., the following rule obtained from step 1:

$$if\ Hand(Tool\_use) \rightarrow Activity(\textbf{Granular}) \tag{4}$$

For this second step, we use as input the activities clustered as *Granular* from the previous step and we learn a new tree, which represents the extension of our previous tree. The methodology that we follow is similar to the one explained in subsection 4.1. It means that the set of instances S is described by similar attributes A, but with different values. For example, the *HandMotion* attribute has now only two possible values: *move* or *not_move*. The attribute *ObjectActedOn* presents the new possible values: *pancake, dough, bread, cheese, electric stove*, etc. Whereas the *ObjectInHand* attribute has 4 possible values: *bottle, spatula, knife* and *plastic wrap*. Note, that

the values of the last attribute are the parental classes of the objects. Some examples of the new *state-value* pairs ($\langle s, c(s)\rangle$) are:

$$\begin{aligned}\langle\{\ Move\ ,\ Pancake\ ,\ Spatula\ \}\ , Slide\_out\rangle\\ \langle\{\ Move\ ,\ \ Bread\ \ ,\ \ Knife\ \}\ ,\ \ \ Cut\ \ \ \rangle\\ \langle\{Not\_Move,\ Cheese,\ Bottle\},\ \ \ Sprinkle\ \ \rangle\end{aligned}$$

4.3 *Knowledge and reasoning engine*

Knowledge and reasoning play a crucial role in dealing with partially observable information. This is possible since they are capable to infer or predict different behaviors, in a way as we (humans) do and expect. This is partly obtained due to the fact that the knowledge base system can combine general knowledge with the current perception of the world to infer hidden aspects of the current state [51].

The Knowledge and Reasoning engine presented in this work uses the Web Ontology Language (OWL), which is an action representation based on logic description as Prolog queries. We use KnowRob [12] as the base line ontology and we incorporate new relationships between objects and actions, additionally we define new activity classes. In order to define *meaningful* relationships between actions and objects, we use the obtained *semantic rules* described in Sections 4.1 and 4.2.

From the obtained rules (see Fig. 6), we implement new *Class Computables*[1] such as *comp_humAct*, to semantically relate the instances from the class *Motion* (*comp_humAct* :′ **Motion**′) with the object properties (*ObjectActedOn* or *ObjectInHand*). The implemented *Computables* are incorporated within our new Prolog predicates using the obtained rules to define new individuals and new relationships between individuals (*objects properties*), as follows:

$$\begin{aligned}&humanAct(?Occ, take) :-\\ &\quad rdfs\_instance\_of(?InstM, comp\_humAct :' \mathbf{Motion}'),\\ &\quad InstM =' StandingStill',\\ &\quad rdf\_triple(comp\_humAct :' \mathbf{objectInHand}', Occ, ?V).\\ &\quad (V =' Something'; V \backslash =' none').\end{aligned} \tag{5}$$

where ?*Occ* is the occurrence number we want to infer and the argument *take* is the name of the inferred class. From the above Prolog predicate we can see how the instances ($?InstM$) of the class *Motion* (*comp_humAct* :′ **Motion**′) and the objects ($?V$) with the property of *ObjectInHand* (*comp_humAct* :′ **objectInHand**′) are semantically described and represented. Similar prolog predicates are defined for the remaining rules. The prolog predicate shown in Eq. (5) is used instead of the *if-then* rule from Algorithm 1. The reader can find more examples regarding the knowledge implementation in [6].

4.4 *Results on the automatic recognition and understanding*

First, we build a decision tree for the *basic* human activities: *reach, take, put, release, idle* and *granular* using the Weka software [52] and the *sandwich making* scenario is chosen as the training data set. This scenario was selected since it represents the highest complexity of the analyzed

[1]Reasoning with *Computables* is another important characteristic of KnowRob [12], since it provides the possibility of compute new relations during the reasoning process (on demand) instead of defining them manually. In this case, the *Computable Class*, creates instances of their target class on demand.

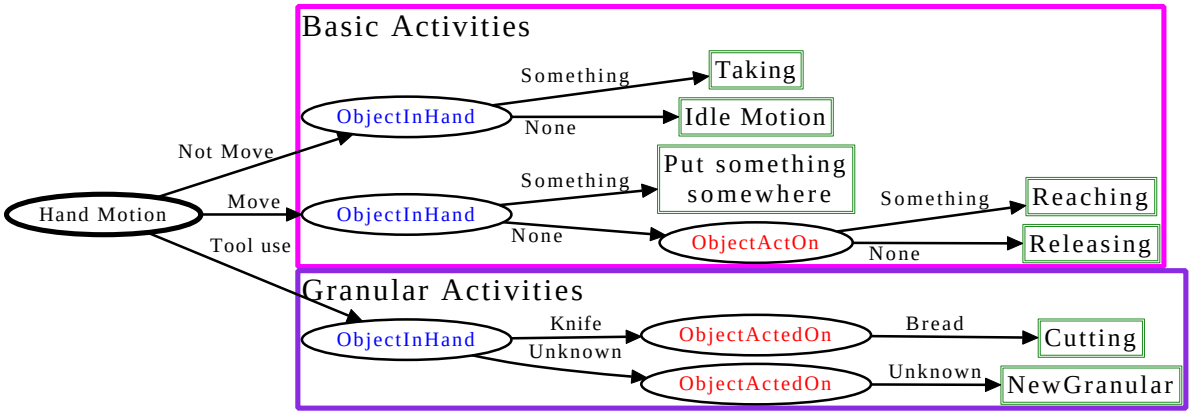


Figure 6. This figure shows on the top part (magenta box) the tree obtained from the sandwich making scenario ($T_{sandwich}$). On the bottom (purple box) is shown the extension of the tree to infer *granular* activities.

tasks due to the several sub-activities that it contains. We use the ground-truth data from subject 1 during a normal speed condition of the *sandwich making* data set. To assess the obtained tree, we use the *10-fold* cross validation option. The obtained tree $T_{sandwich}$ is shown on the top part of Fig. 6 (magenta box). From this tree the recognition accuracy is 92.17%. The obtained tree using our proposed method has 6 leaves (number of rules), i.e. each leave represents one basic human activity: *reach, take, put, release, idle* or *granular*. The size of the tree is only 10 nodes, which means that the complexity of the obtained tree is very low.

For the second step, we use as input the activities clustered as *granular* from the previous step and we apply a similar procedure as before in order to extend our previous tree $T_{sandwich}$. The final tree can be observed in Fig. 6, where the bottom part (purple box) presents the extension of the tree, given the current information of the environment for the task of *cutting the bread*. Notice, that with this methodology the taxonomy of the tree is obtained which allows to add new rules (branches) when a new activity is detected. An example of the implementation of this module is shown in Algorithm 1.

At first glance, steps $1-7$ from Algorithm 1 look like simple $if-then$ rules, however this algorithm is simplified to explain to the reader about the intuitiveness of our proposed method, which is not just a black box as typical reasoners such as Markov Models or Neuronal Networks. The advantage of our system is the possibility to interpret inference errors. For example, if the recognition fails, we can back trace the tree and detect what parameter was incorrectly segmented. This is possible due to the obtained decision tree (see Fig. 6), where these rules can be integrated in any programming language. Nevertheless, if a first-order logic program, e.g. Prolog along with a Knowledge Base is used, as proposed in Section 4.3, then the system is more robust than other semantic approaches [13, 45]. In this case the $if-then$ representation of the activity *take* is expressed in the Prolog predicate shown in Eq. (5).

The next step uses 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 3.3). 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 speed. The average results of both hands show that the accuracy of recognition is about 81.74% (normal condition= 79.18% and fast speed condition=83.43%). The errors in the activity recognition are due to the misclassified objects from the perception module, specially for the sandwich scenario, when the object *knife* is occluded between the hand and the bread (see Fig. 7). One example of the obtained confusion matrix from this scenario is depicted in Fig. 9(a), where the output of the left hand recognition is shown.

Then, we tested the semantic rules into a new scenario (*pouring the pancake mix*), in which the activity *pour* has not yet been learned. Nevertheless, the system is able to identify that a

Algorithm 1 Definition of *getActivity()* algorithm.

Require: m: human motions, e.g. *move, not move or tool use*
o_a : ObjectActedOn property
o_h : ObjectInHand property
memory : memory file that contains new learned activities

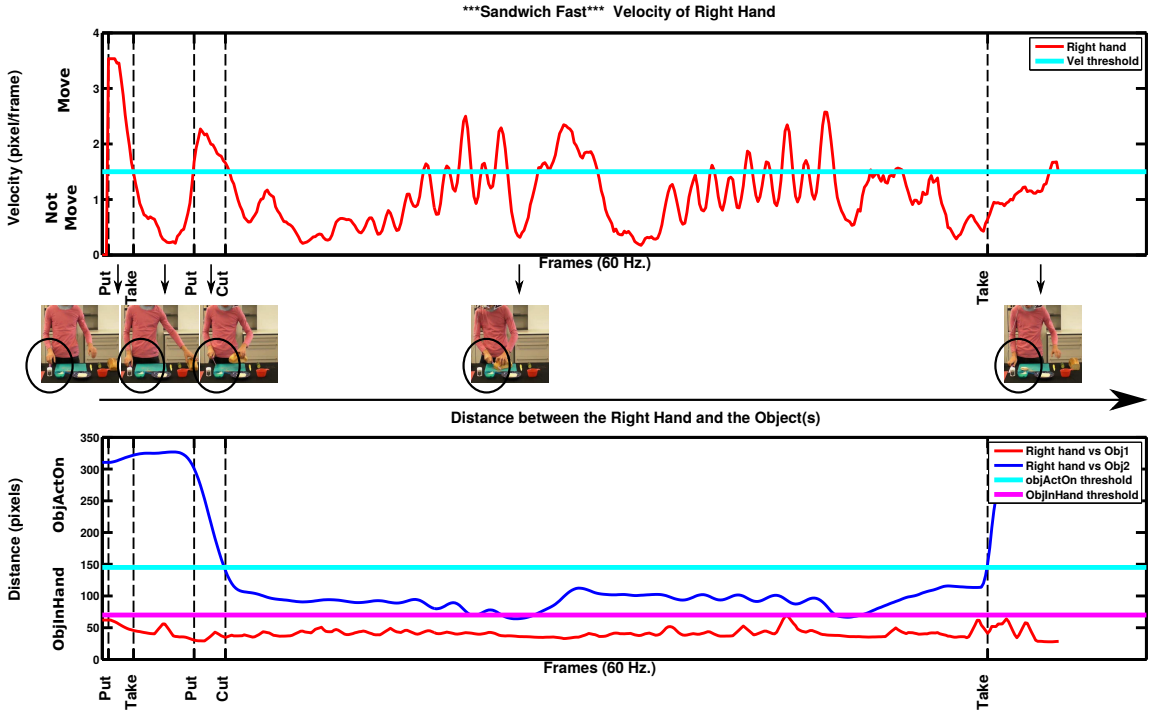
1: **if** $(m ==' not\ move')$ *and* $(o_h ==' none')$ **then**
2: $activity = Idle$
{Notice, that when the system has the knowledge-based enabled, then this *if-then* rule is replaced by its corresponding Prolog predicate similar to Eq. (5).}
3: **else if** $(m ==' ?m')$ *and* $(o_h ==' ?o'_h)$ *and* $(o_a ==' ?o'_a)$ **then**
4: $activity = ?a$ {Replace the content of $?m, ?o_h, ?o_a$ using the corresponding information of the obtained semantic rules shown in Figure 6 (magenta box), where $?a$ could be *take, release, reach or put*}
5: **else if** $(m ==' tool\ use')$ *and* $(o_h ==' knife')$ *and* $(o_a ==' bread')$ **then**
6: $activity = Cut$ {Notice that for the definition of the granular activities shown in Figure 6, we require the context information}
7: **else**
8: $newAct = find\_newActivity(memory, o_h, o_a)$ {A new activity has been detected, e.g. *pour*. Then, first we look into the memory file to find out if the rule has been already learned}
9: **if** $newAct ==' \ '$ **then**
10: $newAct = askUser()$ {If the new activity is not in the memory file, then this function displays a message (during *execution time*) with the identified values of o_h and o_a, then the user is asked to add the name of the new activity (not the rule)}
11: $newRule = createNewRule(newAct, o_h, oa)$ {The system automatically generates the new rule}
12: $memory = saveNewActivity(memory, newRule)$ {The new rule is asserted into the *memory* file similar to step 5 of this algorithm}
13: **else**
14: $activity = newAct$
15: **end if**
16: **end if**
17: $highlight\_branch(activity)$ {Highlight the branch of the tree that corresponds to the inferred activity.}
18: **return** *activity*

new activity has been detected and asks the user to name the *unknown* activity. After the new activity has been learned the system can correctly infer it. The results indicate that the accuracy of recognition is around 88.27% (see Fig. 8). For this new scenario, the obtained confusion matrix of the recognition of the right hand activities is depicted in Fig. 9(b).

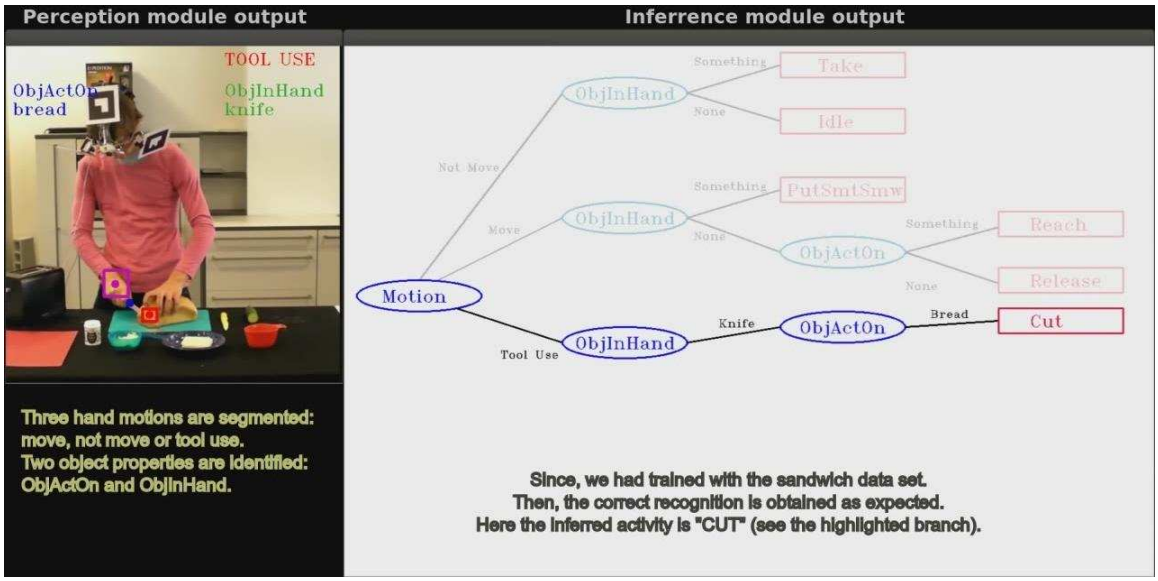
After that, we tested our system with a different task, i.e. *flipping the dough*, similarly to the above example, this demonstration was not trained to recognize the new *granular* activity of *flipping*. The obtained results, shown in the bottom part of Fig. 8, demonstrate that our system is able to learn via *active learning* the new demonstrated activity with an accuracy of 76%. This lower accuracy is due to the errors of the perception system during the identification of the object properties.

Finally, we tested our system with a different scenario *setting the table*. In this case, we use the manually labeled data added with random noise as input. The obtained results suggest that the accuracy of recognition for this scenario is 91.53%. This indicates that our system is capable to recognize human activities without further training for different scenarios using the information acquired only from one demonstration.

The important contribution of these results is the extraction of the semantic rules that allows



(a) Signals from the Color-Base technique



(b) Output of our system to infer *on-line*

Figure 7. These figures show the *on-line* generated signals and the immediate inference of the human activity performed by the right hand of the sandwich scenario when the subjects is in a fast condition. a) shows the signals obtained by the color-based technique where the vertical lines indicate the automatic segmentation and recognition of the human activities for the right hand. Whereas, b) shows one snapshot of our system to infer the human activities *on-line*.

to infer human activities with an overall accuracy of around 85%. The obtained semantic rules were tested for several constraints, such as: demonstration of different activities in different scenarios, where the observed activities were known and unknown. The above is possible even when a very simple hand and object recognition method is used to segment the motions and object properties automatically. Another, very important feature of our system is the possibility of recognizing human activities of both hands at the same time as depicted in Fig. 8. It is possible to observe that the same tree is used to recognize activities for both hands without further modifications.

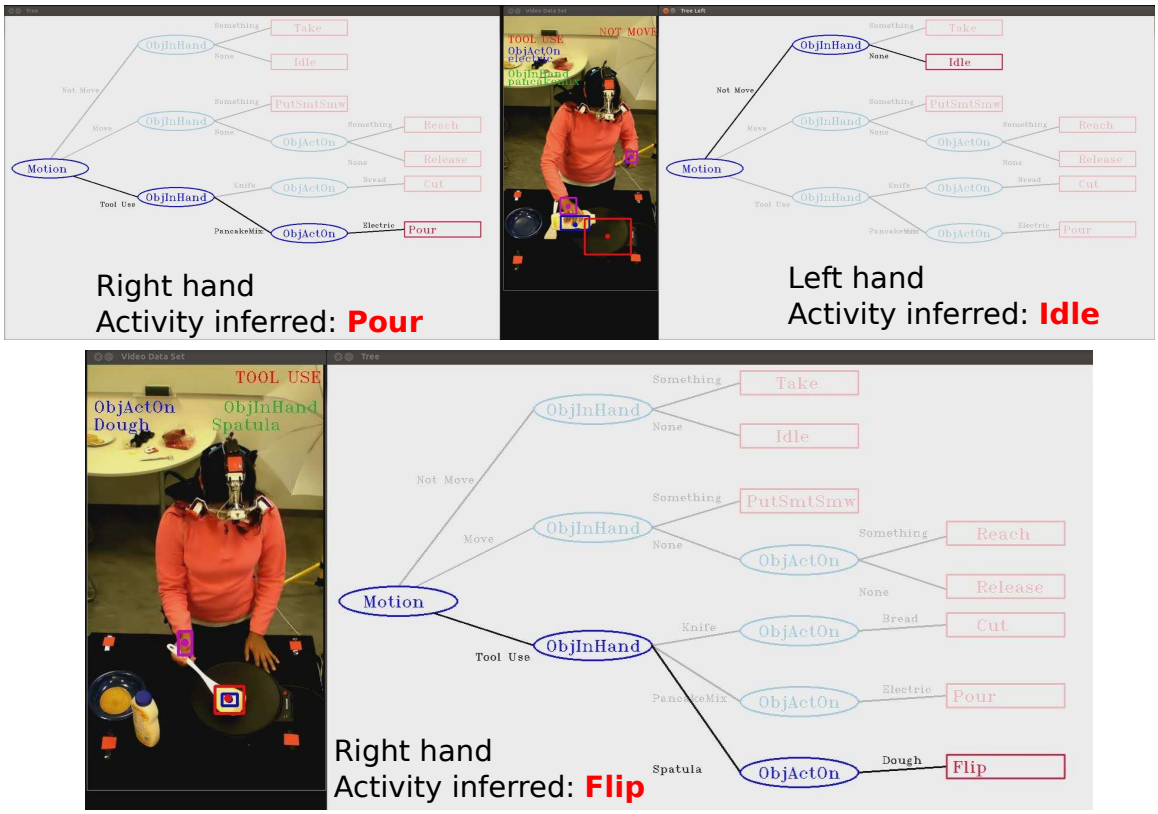
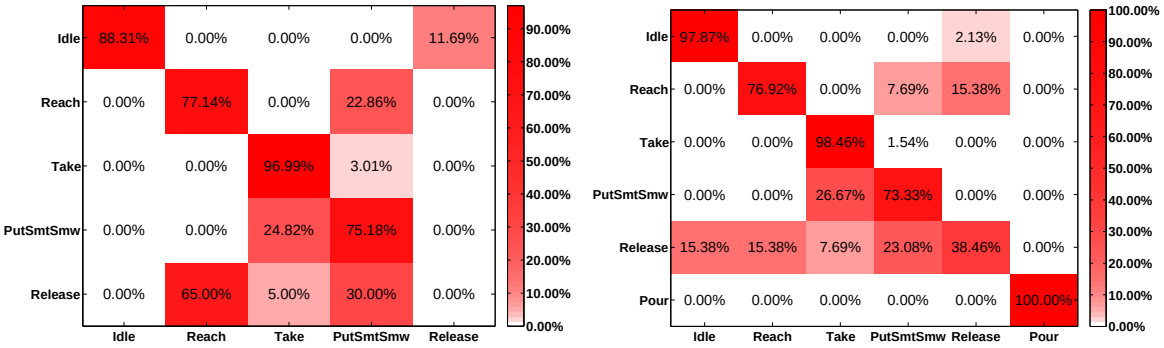


Figure 8. The top part of the figure shows the recognition of both hands at the same time. We observe that the same semantic representation remains for both hands. The bottom part shows the results of a new learned activity *flip* executed by the right hand.



(a) Left hand -*cutting the bread* (b) Right hand -*pouring the pancake mix*

Figure 9. The left figure shows the confusion matrix of the left hand recognition for the scenario *cutting the bread*. The right figure shows the confusion matrix of the new scenario *pouring the pancake mix* for the right hand recognition.

5. Experimental integration on the iCub

The experimental integration and validation of the acquired cognitive behavior into a humanoid robot is very important, essential and a challenging task, which is also addressed in this paper. This represents another key factor of our framework since we integrate the perception and semantic reasoning capabilities to a humanoid robot, in this case the iCub. Our work is not limited to a theoretical domain, but rather, to provide a functional system capable to interact in real scenarios. This integration represents a very challenging task and its solution is not trivial, since it requires the implementation of interfaces between *high-level* control (decision making modules) with the *low-level* control (motion control) to generate a functional system.

For the hardware implementation we used the iCub platform, which consists on 53 degrees of

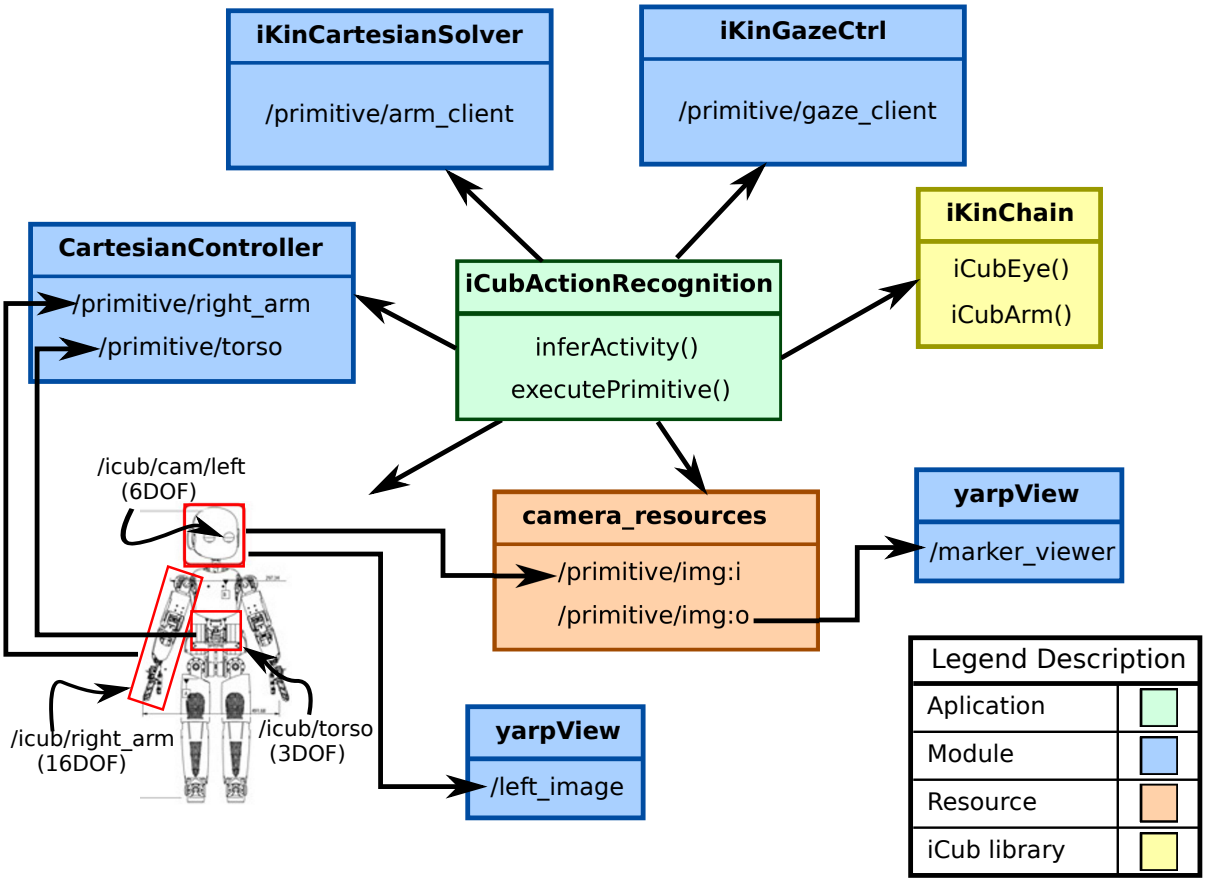


Figure 10. Illustration of all the applications, modules, libraries and the communication between them. We can observe that the main application called *iCubActionRecognition* infers and executes the activity by the iCub.

freedom (DOF) [53]. In this work, we used a total of 25 DOF, i.e. we used 16 DOF of the right arm, 3 DOF of the torso and 6 DOF of the head (see Fig. 10). Regarding the software, we used Yarp [54] and iCub [55] libraries.

Figure 10 depicts a general overview of the implemented Yarp modules, input/output resources, applications and iCub libraries used during the development of our system. From this figure we can observe that our application, i.e. *iCubActionRecognition* (green rectangle), communicates with the iCub through three Yarp modules (blue rectangles), e.g. *CartesianController*, *iKinCartesianSolver* and *iKinGazeCtrl*. Furthermore, our application has communication with the camera interfaces of the iCub (orange rectangle).

5.1 *Description of iCubActionRecognition, our application*

Our application has two main functions: *inferActivity()* and *executePrimitive()* as depicted in Fig. 11. The flow of the data in the control loop of the robot is as follows:

(1) The process *inferActivity()* automatically segments and recognizes human activities in *real-time*. In other words, first we display and pre-process the video stream that shows the desired human activity. The output of this module is the segmented *low-level* motions and object properties. The obtained state of the system (s) is used in the semantic system. which retrieves the inferred activity (g) using Algorithm 1 (see Fig. 12).

(2) The process *executePrimitive()* takes as input the inferred activity which triggers the Skill Planner system. This calls the motion Primitives Library that the robot needs to execute to achieve a similar goal as the one observed. There is a skill plan for each inferred activity. When the skill plan is finished, then the robot waits until the next activity has been

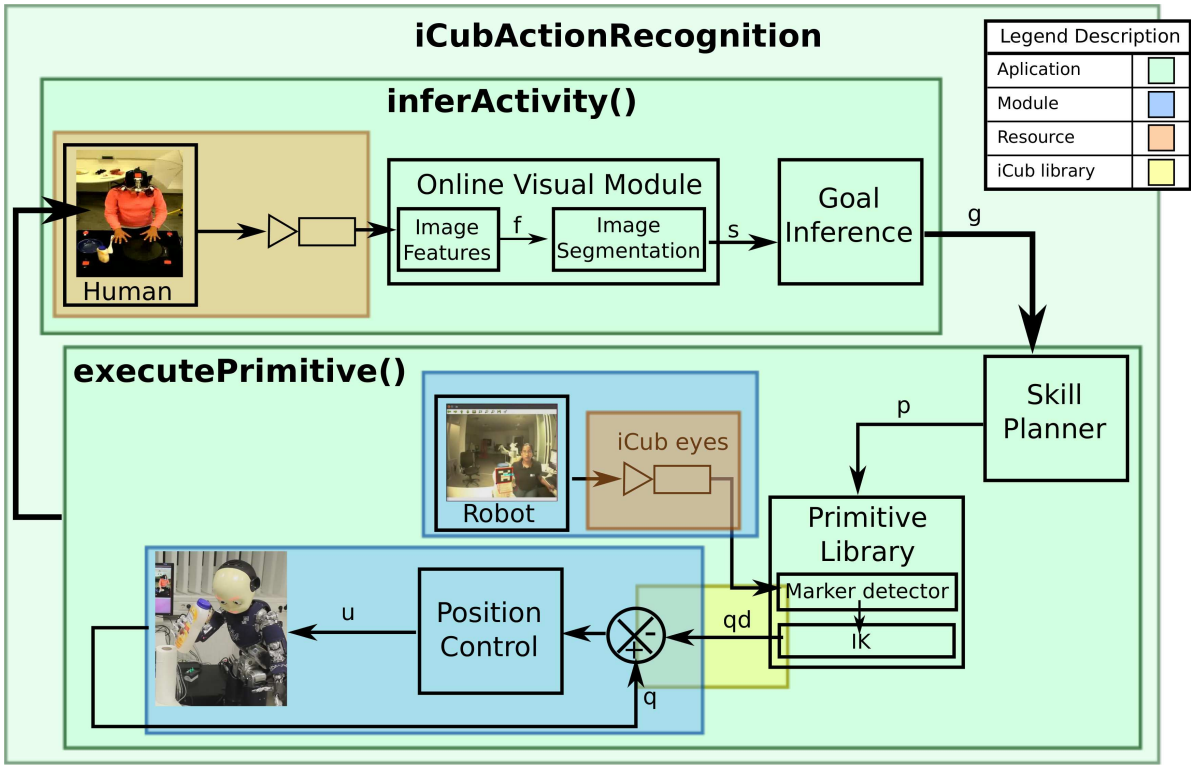


Figure 11. This figure shows the data flow between the processes through Yarp, iCub and other external libraries. We can observe that two principal applications are developed *inferActivity()* and *executePrimitive()*.

inferred. From the execution plan, we obtain $n-$primitives ($p(n)$) that the robot needs to execute. For example, if the inferred activity is *reach*. Then, the function *doReach* is executed (see Fig. 12). When, the execution of *reaching* motion is finished by the robot, then the function *inferActivity()* is executed again to retrieve the next observed human activity. This process is repeated until the information of the video is finished. In the case, that a new activity is detected and learned, for instance the activity of *pouring*, then the function *executePrimitive()* calls the activity *none*, which means that the robot does nothing, unless that activity has been already programmed to the robot.

Notice that all the modules receive inputs and produce desired outputs *on-line*. Thus, with the above processes we have achieved that the robot first observes the human, then it understand the activities performed by the human and finally it execute the corresponding motion.

The robot maps the inferred human activity to a set of preprogrammed motor commands, such as the tilt of the bottle during the pouring activity. These motor commands have been previously preprogrammed due to the fact that it would simplify the determination of the dynamic parameters of the robot, as it can not be obtained easily via observation from the 2D image from our demonstrations, i.e. the dynamics of the person does not match the dynamic of the robot. In this work, we are abstracting the meaning of the observed human motions and we are transferring the obtained models in the robot. This means that at this stage the robot is capable to observe and infer the human activities. After that, the robot uses its own control parameters to execute the action, which represents a complex control problem and some interesting solutions to learn those control parameters have been proposed, e.g. [23] or [**?**].

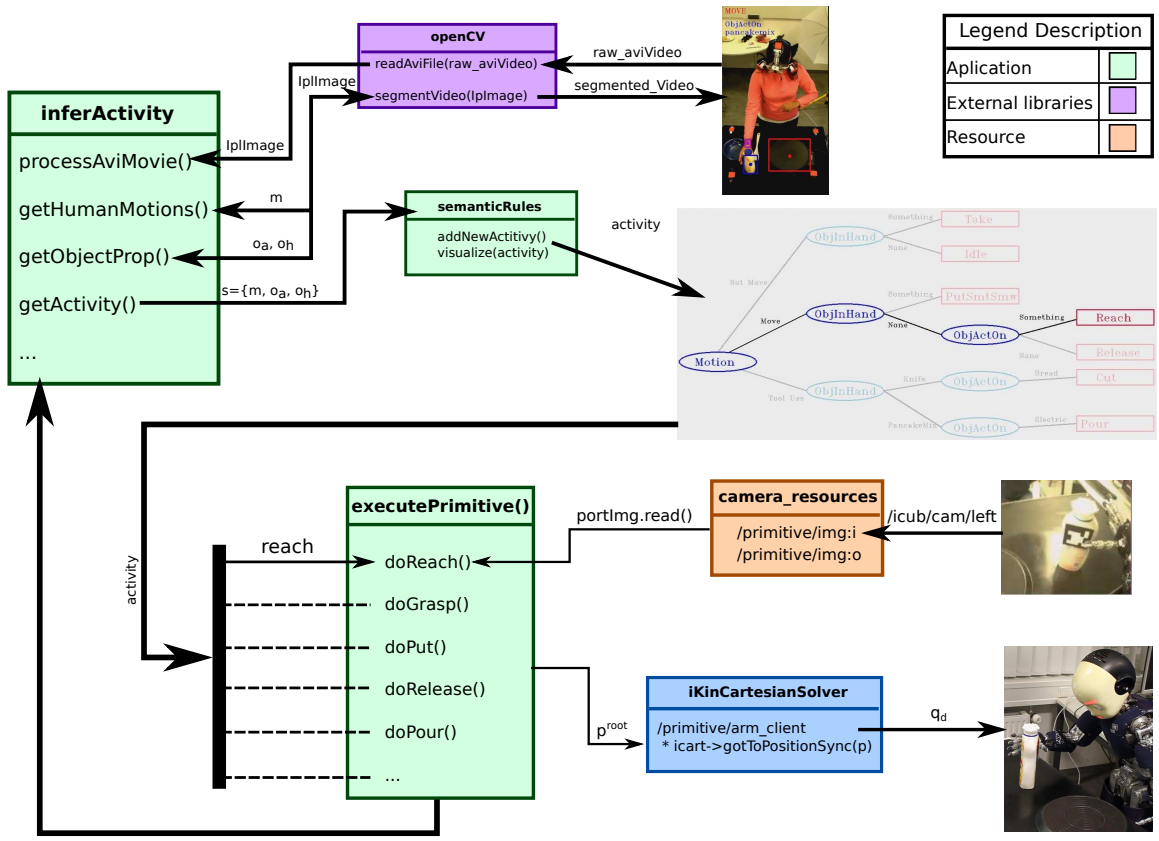


Figure 12. Main applications implemented on the iCub to infer and execute the human activities from observations. The *inferActivity()* application is subdivided in: 1) *processAviMovie()* which applies the OpenCV Color-Based algorithm. 2) *getHumanMotions()*, segments the human motions into: *move, not move* or *tool use*. 3) *getObjectProp()*, obtains the object properties: *ObjectActedOn* and *ObjectInHand*, as presented in Section 3. 4) *getActivity()* infers the human *high-level* activities. The *executePrimitive()* application shows the input/outputs of the programmed robot primitives.

5.2 *Experimental validation on the iCub*

Several experiments were performed to validate and evaluate our work on a humanoid robot in realistic scenarios[1]. To illustrate the different contributions of this work, we show the results of the proposed framework for the *pancake making* scenario as shown in Fig. 13.

In order to evaluate the system response time for observing and inferring, we first analyze the average life time of each of the observed activities in Frames[1] for the three analyzed tasks, i.e, *sandwich making* under normal condition, *sandwich making* under fast condition and *pancake making* (see Table 3). The duration of the videos is different for each task as well as the frequency of the videos. For example, the *sandwich making* under normal condition has a duration of 20 *s* and a frequency of 60 *fps*. Whereas, the *sandwich making* under fast condition and same frequency has a duration of 7 *s*. Finally, the *pancake making* has a duration of 10 *s* with lower frequency 24 *fps*.

Table 3 shows that the shortest activity, i.e. *idle* takes about 7 frames, which indicates the minimum life-time of an activity. A similar analysis in Seconds is shown in Table 4. As expected, these activities differ in time for each task. Noticeable, we observe that the shortest time is 0.12 *s*. Therefore, the robot can make an informative decision after this time to guarantee that the inferred activity has been correctly inferred and executed. Even when the robot is able to infer a new activity each frame, the robot only executes the inferred activity if this has been the

[1]A video where more details for all these experimental results are illustrated can be found in the following link: http://web.ics.ei.tum.de/~karinne/Videos/AR14ramirezK.avi

[1]Notice, that the frame information comes from the manual annotated videos made by a human expert, i.e. from the ground truth data.

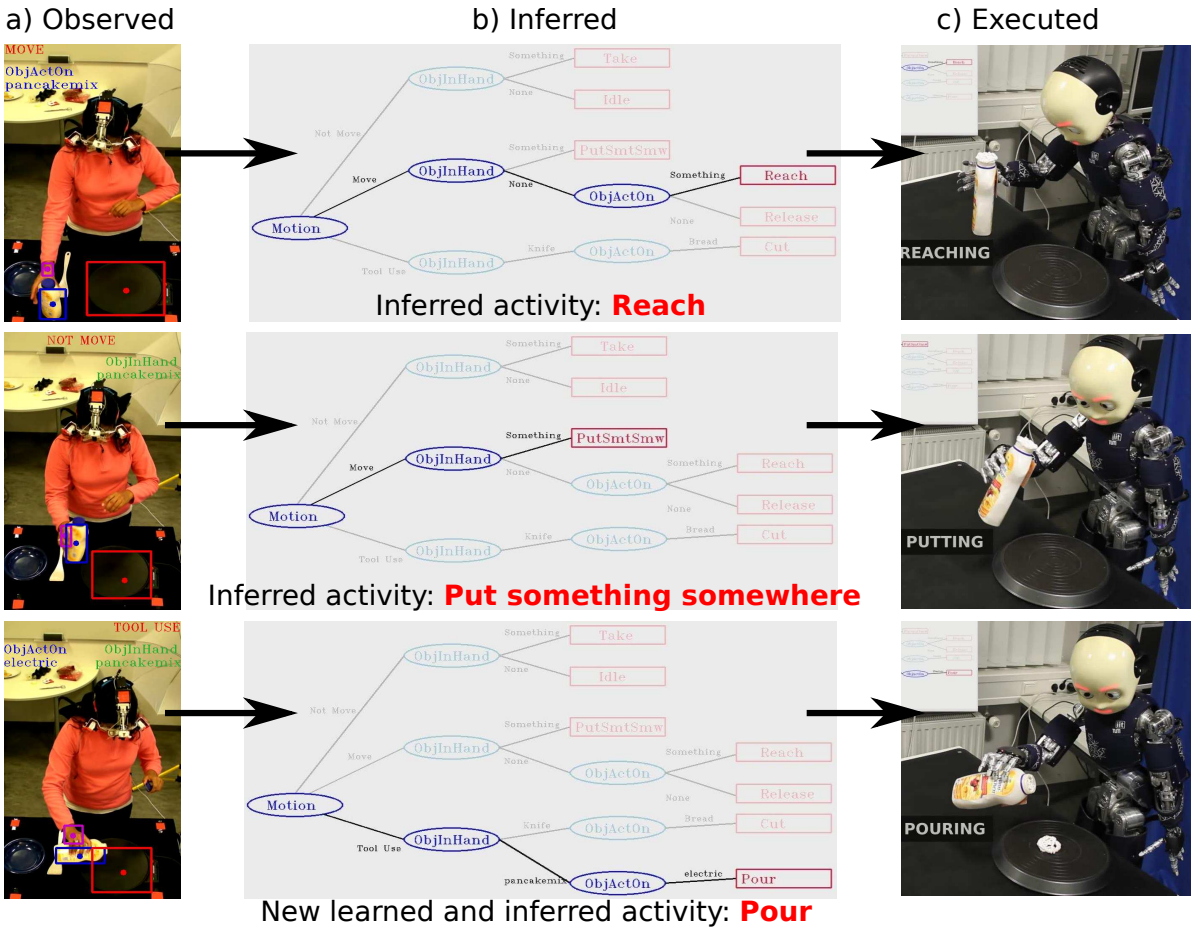


Figure 13. First the robot observes the motions of the human from a video, then it infers or learns the human activity and finally the iCub executes a similar activity.

Table 3. Average activity life time in **Frames**

Activity	Sandwich Normal	Sandwich Fast	Pancake	Inference time of our system
Reach	39	16	10	7
Take	177	68	52	7
Put	61	38	21	7
Release	26	13	15	7
Idle	82	**7**	35	7
Cut	787	280	N/A	7
Pour	N/A	N/A	97	7

same activity for the last 7 frames (i.e. 0.12 s). The reason of this waiting time is to compensate the errors coming from the perception system.

In other words, our proposed semantic-based framework can minimize failures due to occasional frame information loss and/or any incorrectly perceived signals. This can be achieved due to the inference time of 0.12 s (see Table 4) to assure the robot that the observed activity has been correctly inferred. After the inferred activity is extracted, the robot will start the execution of the inferred activity. However, if the robot fails during the execution of the activity, it will simply execute the next inferred activity – in the current system, recovery of such a consequence is not yet supported. Thus, this paper does not address the problem of error recovery[1]. Nevertheless,

[1]The main problem of error recovery in complex manipulation tasks is that the range of failures and undesired outcome of the robot's executions is very large. For example, during the pancake making scenario the possible failures are: the robot pours too much pancake mix, or too little, the robot flips the pancake too soon, or too late, etc.

Table 4. Average activity life time in **Seconds**

Activity	Sandwich Normal	Sandwich Fast	Pancake	Inference time of our system
Reach	0.65	0.27	0.42	0.12
Take	2.95	1.13	2.17	0.12
Put	1.02	0.64	0.88	0.12
Release	0.44	0.22	0.63	0.12
Idle	1.37	**0.12**	1.46	0.12
Cut	13.12	4.66	N/A	0.12
Pour	N/A	N/A	4.04	0.12

since this a very important problem, we are working on adapting our framework to trigger on identified errors during execution utilising the robot sensors and the acquired semantic rules.

When comparing the time performance of our system (0.12 s) with the most recent state-of-the art techniques such as OACs [45], we can observe that our approach infers the observed activity faster than previous approaches. For instance, from the recent video of the OACs concept [45], we determine the inference time of their system. First, we observed that this time is variant since it strongly depends on the life time of the activity, this means that the system needs to observe the effects/consequences of the executed activity. Therefore, the inference time of the OACs approach goes from 0.53 s up to 1 s for this video. Whereas our system can make an inference in 0.12 s for every observed activity. For example the *pouring* activity has an average life-time of 4.04 s, which means that the OACs method will infer this activity at the end of this activity (after 4.04 s), whereas our system can infer this activity in 0.12 s.

6. Conclusions

Understanding human intentions has received significant attention in the last decades since this represents a very important role in Cognitive Systems. 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 by defining two levels of abstraction. The transition between these levels of abstraction is managed by our *semantic* module. Our system contains principally three modules: 1) *low-level* activity observation; 2) interpretation of *high-level* human activities; and 3) the *execution* of the inferred activity by the robot.

The extraction of abstract representations of the observed task, represents a big advantage compare with classical approaches [13] when the task is learned for a specific scenario or a robot. Then, the obtained model only contains the information for that specific task, where the generalization is not possible. Different approaches strongly depend on pre- and post-conditions of the perceived demonstration to recognize human activities e.g. [15, 33, 45]. However, even when these systems are accurate, they need to observe the whole activity (from beginning to end) to correctly recognize it and typically a very sophisticated perception system is needed.

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 tasks. Then, we demonstrated that the inferred representations do not depend on the performed task. Furthermore, the proposed system is able to recognize *new* activities and learn the correct rule(s) *on-line*, which means that we do not need to include all possible activities to the system, since this is not feasible in real applications. This indicates that our system is adaptable and scalable. Noticeable, our approach enable robots to recognize the observed activities in 0.12 s. 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* and by the ICT Call 7 ROBOHOW.COG (FP7-ICT) under grant agreement no. 288533. K. Ramirez-Amaro was supported by a CONACYT-DAAD scholarship.

References

[1] Buss M, Beetz M. CoTeSys – Cognition for Technical Systems. Künstliche Intelligenz. 2010;.
[2] Beetz M, Kirsch A. Special Issue on Cognition for Technical Systems. Künstliche Intelligenz. 2010; 24.
[3] Schaal S. Is imitation learning the route to humanoid robots? Trends in cognitive sciences. 1999; 3(6):233–242.
[4] Nehaniv CL, Dautenhahn K, editors. Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions. Cambridge: Cambridge Univ. Press. 2007.
[5] Xia L, Chen CC, Aggarwal J. Human Detection Using Depth Information by Kinect. In: International Workshop on Human Activity Understanding from 3D Data in conjunction with 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Colorado Springs, USA.. 2011 June.
[6] Ramirez-Amaro K, Beetz M, Cheng G. Extracting Semantic Rules from Human Observations. In: ICRA workshop: Semantics, Identification and Control of Robot-Human-Environment Interaction.. Karlsruhe, Germany.. 2013 May.
[7] Ramirez-Amaro K, Kim ES, Kim J, Zhang BT, Beetz M, Cheng G. Enhancing Human Action Recognition through Spatio-temporal Feature Learning and Semantic Rules. In: Humanoid Robots, 2013, 13th IEEE-RAS International Conference. 2013 October.
[8] Le QV, Zou WY, Yeung SY, Ng AY. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: CVPR. IEEE. 2011. p. 3361–3368.
[9] Ramirez-Amaro K, Beetz M, Cheng G. Automatic Segmentation and Recognition of Human Activities from Observation based on Semantic Reasoning . In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), [Accepted]. IEEE. 2014 Sept.
[10] Poppe R. A survey on vision-based human action recognition. Image Vision Comput. 2010;28(6):976–990.
[11] Aggarwal JK, Ryoo MS. Human activity analysis: A review. ACM Comput Surv. 2011;43(3):16:1–16:43.
[12] Beetz M, Tenorth M, Jain D, Bandouch J. Towards Automated Models of Activities of Daily Life. Technology and Disability. 2010;22.
[13] Park S, Aggarwal J. Semantic-level Understanding of Human Actions and Interactions using Event Hierarchy. In: Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on. 2004 June. p. 12–12.
[14] Aksoy EE, Abramov A, Dörr J, Ning K, Dellen B, Wörgötter F. Learning the semantics of object-action relations by observation. I J Robotic Res. 2011;30(10):1229–1249.
[15] Wörgötter F, Agostini A, Krüger N, Shylo N, Porr B. Cognitive agents - a procedural perspective relying on the predictability of Object-Action-Complexes (OACs). Robotics and Autonomous Systems. 2009;57(4):420–432.
[16] Vernon D, Metta G, Sandini G. A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents. IEEE Trans Evolutionary Computation. 2007;11(2):151–180.
[17] Kuniyoshi Y, Inoue H. Qualitative Recognition of Ongoing Human Action Sequences. In: Bajcsy R, editor. IJCAI. Morgan Kaufmann. 1993. p. 1600–1609.
[18] Inamura T, Shibata T. Geometric Proto-Symbol Manipulation towards Language-based Motion Pattern Synthesis and Recognition . In: International Conference on Intelligent Robots and Systems (IROS) 2008. IEEE. 2008. p. 334–339.
[19] Billard A, Calinon S, Dillmann R, Schaal S. Survey: Robot Programming by Demonstration. Hand-

book of Robotics. 2008.
[20] Schuldt C, Laptev I, Caputo B. Recognizing human actions: a local SVM approach. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. Vol. 3. IEEE. 2004. p. 32–36.
[21] Albrecht S, Ramirez-Amaro K, Ruiz-Ugalde F, Weikersdorfer D, Leibold M, Ulbrich M, Beetz M. Imitating human reaching motions using physically inspired optimization principles. In: Humanoids. IEEE. 2011. p. 602–607.
[22] Nehaniv C, Dautenhahn K. The correspondence problem. In: Dautenhahn K, Nehaniv C, editors. Imitation in animals and artifacts. MIT Press. 2002. p. 41–61.
[23] Bentivegna DC, Atkeson CG, Cheng G. Learning Similar Tasks From Observation and Practice. In: IROS. IEEE. 2006. p. 2677–2683.
[24] Billard A, Epars Y, Calinon S, Cheng G, Schaal S. Discovering Optimal Imitation Strategies. Robotics and Autonomous System. 2004;47(2–3):67–77.
[25] Calinon S, Billard A. Incremental learning of gestures by imitation in a humanoid robot. In: ACM/IEEE international conference on Human-robot interaction. ACM. 2007. p. 255–262.
[26] Ude A, Gams A, Asfour T, Morimoto J. Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives. IEEE Transactions on Robotics. 2010;26(5):800–815.
[27] Ijspeert AJ, Nakanishi J, Schaal S. Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots. In: ICRA. IEEE. 2002. p. 1398–1403.
[28] Atkeson C, Schaal S. Robot learning from demonstration. In: Proc. Int. Conf. on Machine Learning. 1997. p. 12–20.
[29] Takano W, Nakamura Y. Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation. In: Humanoid Robots, 2006 6th IEEE-RAS International Conference on. IEEE. 2006. p. 425–431.
[30] Ikeuchi K, Suchiro T. Towards an assembly plan from observation. I. Assembly task recognition using face-contact relations (polyhedral objects). In: Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on. 1992 May. p. 2171–2177 vol.3.
[31] Kuniyoshi Y, Inaba M, Inoue H. Learning by watching : Extracting reusable task knowledge from visual observation of human performance. IEEE Transactions on Robotics and Automation. 1994; 10(6):799–822.
[32] Ogawara K, Tanuki T, Kimura H, Ikeuchi K. Acquiring Hand-action Models by Attention Point Analysis. In: ICRA. IEEE. 2001. p. 465–470.
[33] Jäkel R, Schmidt-Rohr SR, Lösch M, Dillmann R. Representation and constrained planning of manipulation strategies in the context of Programming by Demonstration. In: ICRA. IEEE. 2010. p. 162–169.
[34] Fern A, Siskind JM, Givan R. Learning Temporal, Relational, Force-Dynamic Event Definitions from Video. In: Dechter R, Sutton RS, editors. AAAI/IAAI. AAAI Press / The MIT Press. 2002. p. 159–166.
[35] Turaga PK, Chellappa R, Subrahmanian VS, Udrea O. Machine Recognition of Human Activities: A Survey. IEEE Trans Circuits Syst Video Techn. 2008;18(11):1473–1488.
[36] Gupta A, Davis L. Objects in action: An approach for com- bining action understanding and object perception . In: IEEE Conference in Computer Vision and Pattern Recognition . IEEE. 2007. p. 1–8.
[37] Ghanem N, Dementhon D, Doermann D, Davis L. Representation and recognition of events in surveillance video using Petri nets. In: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW) . Los Alamitos, CA.. IEEE. 2004.
[38] Kitani KM, Sato Y, Sugimoto A. Recovering the Basic Structure of Human Activities from Noisy Video-Based Symbol Strings. IJPRAI. 2008;22(8):1621–1646.
[39] Ivanov YA, Bobick AF. Recognition of Visual Activities and Interactions by Stochastic Parsing. IEEE Trans Pattern Anal Mach Intell. 2000;22(8):852–872.
[40] Siskind JM. Grounding the Lexical Semantics of Verbs in Visual Perception using Force Dynamics and Event Logic. J Artif Intell Res (JAIR). 2001;15:31–90.
[41] François ARJ, Nevatia R, Hobbs JR, Bolles RC. VERL: An Ontology Framework for Representing and Annotating Video Events. IEEE MultiMedia. 2005;12(4):76–86.
[42] Sridhar M, Cohn AG, Hogg DC. Learning Functional Object-Categories from a Relational Spatio-Temporal Representation. In: Ghallab M, Spyropoulos CD, Fakotakis N, Avouris NM, editors. ECAI. Vol. 178 of Frontiers in Artificial Intelligence and Applications. IOS Press. 2008. p. 606–610.
[43] Yang Y, Fermüller C, Aloimonos Y. Detection of Manipulation Action Consequences (MAC). In:

CVPR. IEEE. 2013. p. 2563–2570.
[44] Guha A, Yang Y, Fermuuller C, Aloimonos Y. Minimalist plans for interpreting manipulation actions. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. 2013. p. 5908–5914.
[45] Wächter M, Schulz S, Asfour T, Aksoy E, Wörgötter F, Dillmann R. Action Sequence Reproduction based on Automatic Segmentation and Object-Action Complexes. In: IEEE/RAS International Conference on Humanoid Robots (Humanoids). 2013.
[46] Kjellström H, Romero J, Kragic D. Visual object-action recognition: Inferring object affordances from human demonstration. Computer Vision and Image Understanding. 2011;115(1):81–90.
[47] Bradski GR, Kaehler A. Learning OpenCV - computer vision with the OpenCV library: software that sees. O'Reilly. 2008.
[48] Ramirez-Amaro K, Chimal-Eguia J. Image-Based Learning Approach Applied to Time Series Forecasting. Journal of Applied Research and Technology. 2012 June;10(3):361–379.
[49] Tenorth M, Beetz M. KnowRob: A knowledge processing infrastructure for cognition-enabled robots. I J Robotic Res. 2013;32(5):566–590.
[50] Quinlan R. C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann Publishers. 1993.
[51] Russell SJ, norving P. Artificial Intelligence: A Modern Approach. Upper Saddle River, New Jersey: Prentice-Hall. 1995.
[52] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. SIGKDD Explor Newsl. 2009;11(1):10–18.
[53] Metta G, Sandini G, Vernon D, Natale L, Nori F. The iCub humanoid robot: an open platform for research in embodied cognition. In: PerMIS: Performance Metrics for Intelligent Systems Workshop. 2008. p. 19–21.
[54] Metta G, Fitzpatrick P, Natale L. YARP: Yet Another Robot Platform. International Journal of Advanced Robotics Systems, special issue on Software Development and Integration in Robotics. 2006;3(1). Available from: http://www.bibsonomy.org/bibtex/2f0bb53290115f13f411351a3ec17dfd6/bernuly.
[55] Pattacini U. Modular Cartesian Controllers for Humanoid Robots: Design and Implementation on the iCub. [Ph.D. thesis]. 2010.

Biographies

Karinne Ramirez Amaro is a PhD researcher since January 2013 at the Institute for Cognitive Systems from the Electrical Engineering Faculty, Technische Universität München under the supervision of Prof. Gordon Cheng. From October 2009 until Dec 2012, she was a member of the Intelligent Autonomous Systems (IAS) group leaded by Prof. Michael Beetz. She received a Master in Computer Science with honors at the Center for Computing Research of the National Polytechnic Institute (CIC-IPN) in Mexico City, Mexico in November 2007. She studied Computer Systems Engineering at the Technological Institute of Merida (I.T.M.) in Merida, Yucatan, Mexico, where she received her Bachelor degree in May 2004. She was awarded with a scholarship for a Ph. D. research by DAAD - CONACYT and She received the Google Anita Borg scholarship in 2011. Her research interests include semantics representations in order to understand and to find models that generalize and explain the human everyday activities from observations.

Michael Beetz is a professor for Computer Science at the Faculty for Informatics at the University of Bremen and head of the Institute for Artificial Intelligence. From 2006 to 2011, he was vice coordinator of the German national cluster of excellence CoTeSys (Cognition for Technical Systems). Michael Beetz received his diploma degree in Computer Science with distinction from the University of Kaiserslautern. He received his MSc, MPhil, and PhD degrees from Yale University in 1993, 1994, and 1996 and his Venia Legendi from the University of Bonn in 2000. Michael Beetz was a member of the steering committee of the European network of excellence in AI planning (PLANET) and coordinating the research area "robot planning". He is associate editor of the AI Journal. His research interests include plan-based control of robotic agents, knowledge processing and representation for robots, integrated robot learning, and cognitive perception.

Gordon Cheng is the Professor and Chair of Cognitive Systems, and founding Director of the Institute for Cognitive Systems, Technische Universität München, Munich, Germany. He was the Head of the Department of Humanoid Robotics and Computational Neuroscience, ATR Computational Neuroscience Laboratories, Kyoto, Japan, from 2002 to 2008. He was the Group Leader for the JST International Cooperative Research Project, Computational Brain, from 2004 to 2008. He was designated a Project Leader from 2007 to 2008 for the National Institute of Information and Communications Technology of Japan. He has held visiting professorships worldwide in multidisciplinary fields comprising mechatronics in France, neuroengineering in Brazil, and computer science in the USA. He held fellowships from the Center of Excellence and the Science and Technology Agency of Japan. Both of these fellowships were taken at the Humanoid Interaction Laboratory, Intelligent Systems Division at the Electrotechnical Laboratory, Japan. He received the Ph.D. degree in systems engineering from the Department of Systems Engineering, The Australian National University, in 2001, and the bachelors and masters degrees in computer science from the University of Wollongong, Wollongong, Australia, in 1991 and 1993, respectively. He was the Managing Director of the company G.T.I. Computing in Australia. His current research interests include humanoid robotics, cognitive systems, brain machine interfaces, biomimetic of human vision, human-robot interaction, active vision, and mobile robot navigation. He is the co-inventor of approximately 15 patents and has co-authored approximately 250 technical publications, proceedings, editorials, and book chapters.