A Compositional Approach for 3D Arm-Hand Action Recognition

Ilaria Gori[1], Sean Ryan Fanello[1], Francesca Odone[2] and Giorgio Metta[1]

Abstract— **In this paper we propose a fast and reliable vision-based framework for 3D arm-hand action modelling, learning and recognition in human-robot interaction scenarios. The architecture consists of a compositional model that divides the arm-hand action recognition problem into three levels. The bottom level is based on a simple but sufficiently accurate algorithm for the computation of the scene flow. The middle level serves to classify action primitives through descriptors obtained from 3D Histogram of Flow (3D-HOF); we further apply a sparse coding (SC) algorithm to deal with noise. Action Primitives are then modelled and classified by linear Support Vector Machines (SVMs), and we propose an on-line algorithm to cope with the real-time recognition of primitive sequences. The top level system synthesises combinations of primitives by means of a syntactic approach. In summary the main contribution of the paper is an incremental method for 3D arm-hand behaviour modelling and recognition, fully implemented and tested on the iCub robot, allowing it to learn new actions after a single demonstration.**

I. Introduction

Learning and recognition of human actions has become a challenging research topic not only in the computer vision community, but also in the robotics one, due to the recent advances in the development of humanoid robots that enable the recognition of actions in real scenarios such as human-robot interaction (HRI) and learning by imitation settings. Whereas vision-based action recognition systems usually resort to discriminative models gaining computational efficiency and flexibility, robotics favours generative models of actions in order to enable the learning by imitation [1]. In this paper, we propose an approach trying to benefit from both the trends: in fact, our vision-based framework is a compositional system that uses discriminative methods to learn and recognise low-level gestures, whilst high-level actions are caught via a generative approach built on regular languages.

Recent literature is rich of various algorithms for gesture, action, and activity recognition, we refer the reader to [2], [3], [4] for a complete overview of the topic. From these reviews we can infer that a gesture is a low-level component with no a priori semantic meaning, whereas each arm-hand action embodies significant information, and it may be composed of multiple gestures organised as a sequence. Even though gestures show to have sufficient expressive power for specific applications, in other contexts, such as robotics or social interactions, the association of a particular high level meaning to one action is often needed. This motivates the focus on the much more challenging field of action modelling and recognition. The first concern to address the choice of the most suitable representation: we need to design a descriptor with the maximum possible discriminative power. Many well-known approaches rely on descriptors based on temporal templates [5], Histograms of Flow (HOFs) [6], Spatial-Temporal Interest Points [7] or a combination of multiple features [8]. Instead we aim to encode directions in 3D space, which appear to be excellent candidates to represent the primitive components of actions. To this end we develop a simple and efficient scene flow algorithm that is functional to classify actions on the basis of the 3D flow directions. Subsequently we employ multidimensional histograms (3D-HOF), and we apply a sparse coding (SC) method [9] in order to filter noise.

As regards the recognition stage, many techniques use complex structures such as Hidden Markov Models [10], Coupled Hidden Semi-Markov Models [11], or Action Graphs [12]; although the mentioned approaches proved to be effective in many contexts, they require an expensive off-line training procedure in order to learn a new action. Conversely, our purpose is to model an arbitrary number of actions, requiring a fast learning phase, hence we develop a syntactic approach according to Aggarwal's hierarchy [2]: we define a certain number of primitives, which are modelled off-line, and then we combine them in more complex actions that can be learned on-line via one shot demonstration.

More specifically, for the rest of the paper, we define an **action primitive** as a simple arm-hand action that is directly mapped to a motor capability of a robot, e.g. $grasp$. For learning and classification of the primitives we use simple linear SVMs, that ensure constant complexity in testing. For the recognition of a sequence of primitives we analyse the SVMs' classification score trends. These sequences are then represented as words, as they combine primitives to obtain **complex actions**, which belong to the class of regular languages; the recognition of complex actions involves a similarity function between the input string and those generated by the previously learned Finite State Automata.

The proposed method shows excellent performances in real-time while retaining high recognition performances. Besides, our results highlight that 3D-HOF descriptors are suitable for discriminating primitives with high accuracy. Additionally, once primitives have been modelled off-line, it is possible to learn complex actions in real-time by a one shot learning procedure. The paper is organised as follows: in Sec. II

Research supported by the European FP7 ICT project No. 270490 (EFAA) and project No. 270273 (Xperience).

[1]I. Gori, S.R. Fanello and G. Metta are with the Robotics, Brain and Cognitive Sciences Department, Istituto Italiano di Tecnologia, Genova, Italy {ilaria.gori, sean.fanello, giorgio.metta} at iit.it

[2]F. Odone is with the Department of Informatics Bioengineering Robotics and Systems Engineering, Università degli Studi di Genova, Genova, Italy francesca.odone at unige.it

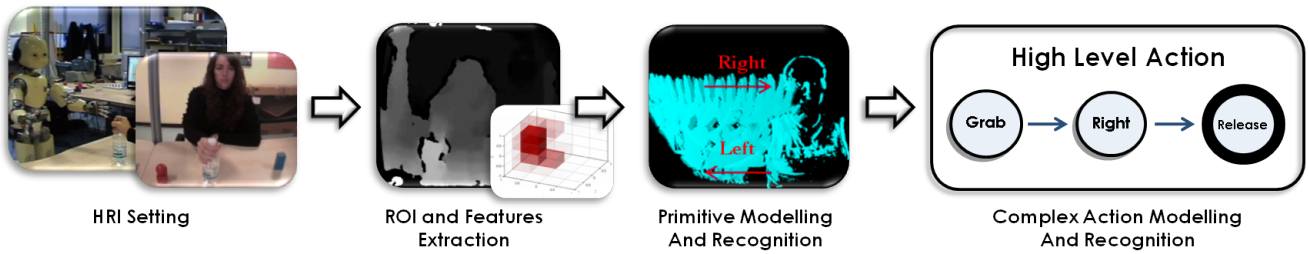

Fig. 1. Overview of the recognition system. From a typical HRI setting we developed a complete framework based on the 3D vision of the iCub, modelling primitive actions via linear SVMs and describing complex actions using regular languages.

we present the overview of the system. Sec. III describes the experiments and corresponding results substantiating our claims. Finally Sec. IV discusses future directions and possible improvements to the current implementation.

II. System Overview

Our system can be thought of as a cascade of three subsequent layers (see Fig. 1):

- **ROI and Features Extraction**: the scene is first segmented from depth information along with motion cues, to identify the subject of interest willing to interact with the robot; subsequently scene flow is computed. Each frame is represented by a multidimensional histogram of the scene flow (3D-HOF), enhanced by a postprocessing via sparse coding techniques.
- **Action Primitive Modelling and Recognition**: a linear SVM on a sliding window of *n* frames is used to classify the current action primitive. Sequences of primitives are recognised by analysing the variation of the classifiers output's trend, and each sequence is represented by a string.
- **Complex Action Modelling and Recognition**: each class of complex actions is described by a Finite State Machine (FSM). The complex action recognition procedure is based on the Levenshtein distance [13] between the recognised string and all the generated strings belonging to the previously learned Finite State Automata.

Our architecture is simple, efficient and fast. In the following we explain these layers in detail.

A. Region of Interest & Features Extraction

Since our work is developed in the context of HRI, it is reasonable to assume that the actor stands in front of the humanoid robot. In our setting the robot mounts two cameras, therefore relying on a stereo vision system – after a proper calibration procedure – appears to be a natural choice. In order to separate interesting objects from the potentially cluttered background we adopt a segmentation method based on depth combined with motion information. Furthermore, since we are mainly interested in interpreting motion, we define features based on a simple scene flow algorithm, designed to project 2D optical flow vectors in the 3D space on the basis of the stereo vision system. In order to make this information manageable a more compact representation is employed, and it is based on 3D Histograms of Flow (3D-HOFs). Finally, in order to reduce the impact of noise, a sparse coding technique is used.

1) Region of Interest Extraction: an essential requirement of every action recognition system is the ability to reliably isolate interesting regions of the image from the background. Many approaches have been proposed, most of them relying on more or less sophisticated background modelling techniques [14], or space-time image filtering in order to extract only interesting spatio-temporal locations of the action [15]. In other cases they require a robust knowledge of the body pose [7]. Our field of interest is human-robot interaction, thus it is reasonable to assume that the subject stands in front of the robot and can be observed by the robot's cameras; for this reason we exploit a simple but efficient segmentation technique, based on the depth map, like [16]. The computation of the depth map is carried out following the method in [17], which allows segmenting the subject from the background reliably (see Fig. 2). Another significant cue is the motion, as the very idea of action is in fact strictly related to the concept of movement. Motion also represents a strong cue to draw the robot's attention to relevant locations in its surrounding, in fact we only retrieve information from those parts of the scene where motion is detected. In the following we thus assume, without loss of generality, that a Region of Interest (ROI) is identified via depth and motion cues.

2) Histogram of Scene Flow: we are interested in features that capture 3D arm-hand actions in space. In order to obtain a feature-like representation of all the possible arm-hand movements a 3D optical flow (namely scene flow) is needed. Whereas for 2D motion vectors estimation a lot of methods have been proposed in the last decades [18], [19], the scene flow computation is still an active research field since its processing implies an additional disparity estimation problem. The most effective algorithms [20], [21] run at 5Hz on CPU assuming the disparity map computed on a dedicated device. Due to the real-time constraints of human-robot interaction and the limited resources shared among the other tasks (segmentation, features computation, classification stage) we moved on a less accurate but more efficient method for the computation of the scene flow. For each frame F_t we compute the 2D flow vectors $U(x, y, t)$

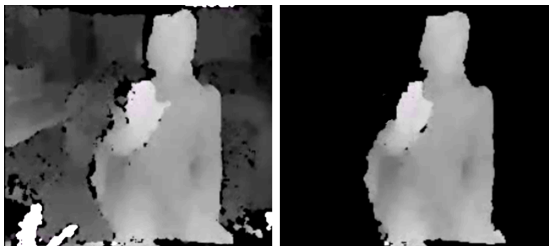

Fig. 2. Region of Interest via depth map computed with [17]. We assume the subject be in front of the robot while he is performing actions.

and $V(x, y, t)$ for the x and y components, with respect to the previous frame F_{t-1} via the Fanerbäck's algorithm [22]. Each pixel (x_{t-1}, y_{t-1}) belonging to the ROI of the frame F_{t-1} is reprojected in the 3D space using the depth map:

$$\begin{pmatrix} X_{t-1} \\ Y_{t-1} \\ Z_{t-1} \end{pmatrix} = \begin{pmatrix} \frac{(x - x_0)b}{d} \\ \frac{(y - y_0)b}{d} \\ \frac{bf}{d} \end{pmatrix} \tag{1}$$

where b is the baseline of the stereo system, f the focal length d is the disparity and $(x_0, y_0)^T$ is the principal point of the stereo camera system. Similarly we can reproject the final point (x_t, y_t) of the 2D vector representing the flow, obtaining another 3D vector $(X_t, Y_t, Z_t)^T$. For each pixel of the ROI, we can define the scene flow as the difference of the two 3D vectors in two successive frames F_{t-1} and F_t:

$$\begin{aligned} \mathbf{D} &= (\dot{X}, \dot{Y}, \dot{Z})^T = \\ &= (X_t - X_{t-1}, Y_t - Y_{t-1}, Z_t - Z_{t-1})^T \end{aligned} \tag{2}$$

Once the 3D flow for each pixel of the ROI has been computed we normalise it with respect to the $L2$-norm, so that the resulting descriptors are invariant to the overall speed of the action. At this stage the frame F_t is represented by a set of 3D motion unit vectors $\mathbf{D_1}, \ldots, \mathbf{D_n}$; in order to obtain a compact representation of the scene flow for each frame we build a 3D Histogram of Flow (3D-HOF). In particular each frame is encoded with the descriptor $\mathbf{z}(t) \in \mathbb{R}^{h \times h \times h}$ where h is the discretisation step parameter of the space (i.e. the bin size). In addition we normalise each 3D-HOF $\mathbf{z}(t)$ (i.e. $\sum_j z(t)_j = 1$) so that it captures the ratio of directions in the current action and descriptors are invariant to scale. This simple scene flow computation algorithm, although less accurate than others in terms of actual pixel displacement, identifies efficiently and effectively the directions generated by the moving object (or arm-hand), while ensuring real-time performances (see Fig. 6).

3) 3D-HOF Sparse Codes: we defined features that can effectively describe the 3D movements of an object or hand. Despite its simplicity, the 3D-HOF descriptor is particularly appropriate to represent a single primitive. It is likely though that this estimation is affected by noise, and it is desirable to filter it beforehand; one of the possible techniques that has been explored recently by the ML community is the so-called sparse coding [9], which have become very popular for many applications such as denoising [23] or object recognition [24]. The main concept behind sparse coding is to compress information in a representation with a greater expressive power; it may be obtained by decomposing the signals into a linear combination of a few elements from a given or learned dictionary. Sparse coding approaches often involve a preliminary stage, called Dictionary Learning, where also the atoms (i.e. elements of the dictionary) are learned directly from the data. Specifically, given the set of the previously computed 3D-HOFs $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n]$ (for clarity, we suppress the explicit dependence on t in the notation), the goal is to learn a dictionary $\mathbf{D}$ and a code $\mathbf{U}$ that minimise the reconstruction error:

$$\min_{\mathbf{D},\mathbf{U}} \|\mathbf{Z} - \mathbf{D}\mathbf{U}\|_F^2 + \lambda \|\mathbf{U}\|_1 \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius norm. Fixing $\mathbf{U}$, the above optimization reduces to a least square problem, whilst, given $\mathbf{D}$, it is equivalent to a linear regression with $L1$ norm. The solution is computed through the *feature-sign search* algorithm [9]. Therefore, after the Features Extraction stage, we can describe a set of n frames as a sequence of codes: $\mathbf{u}_1, \ldots, \mathbf{u}_n$ rather than the original descriptors.

B. Primitive Modelling & Recognition

Each single primitive can be easily learned and classified using standard ML methods; in the following we will detail the primitive learning and recognition procedure, also proposing an online algorithm in order to recognise sequences of primitives.

1) Modelling & Classification of a Single Primitive: action primitives are modelled with one of the classic discriminative machine learning methods, SVM [25]. For each video V_i of t_i frames, containing only one primitive A_s, we compute a set of features $[\mathbf{u_1}, \ldots, \mathbf{u_{t_i}}]$ as described in Sec. II-A.2 and II-A.3. A sliding buffer $\mathbf{B}_T(t)$ concatenating all the frame descriptors from $\mathbf{u_{t-T}}$ to $\mathbf{u_t}$ becomes the descriptor for the classifier at time t. In particular we define the new frame representation $\mathbf{B}_T(t)$ as

$$\mathbf{B}_T(t) = (\mathbf{u_{t-T}}, \ldots, \mathbf{u_{t-1}}, \mathbf{u_t})^T \tag{4}$$

The set of buffers

$$\{\mathbf{B_{is}} = [\mathbf{B}_T(t_0), \ldots, \mathbf{B}_T(t_i)] \ : \ i = 1, \ldots, n\} \tag{5}$$

computed from all the videos $\{V_i : i = 1, \ldots, n\}$ of the class A_s are used as positive examples for the primitive A_s. Given the training data $\{\mathbf{B}, \mathbf{y}\}$ where $\mathbf{B}$ is the set of positive and negative examples for the primitive A_s, $y_i = 1$ if the example is positive, $y_i = -1$ otherwise, the goal of SVM is to learn a linear function $(\mathbf{w}^T, b)$ such that a new test vector $\bar{\mathbf{B}}$ is predicted as:

$$f(\bar{\mathbf{B}}) = y = \mathbf{w}^{\mathrm{T}} \bar{\mathbf{B}} + b \tag{6}$$

We use a one-versus-all strategy to train a binary linear SVM for each primitive A_s, so that at the end of the

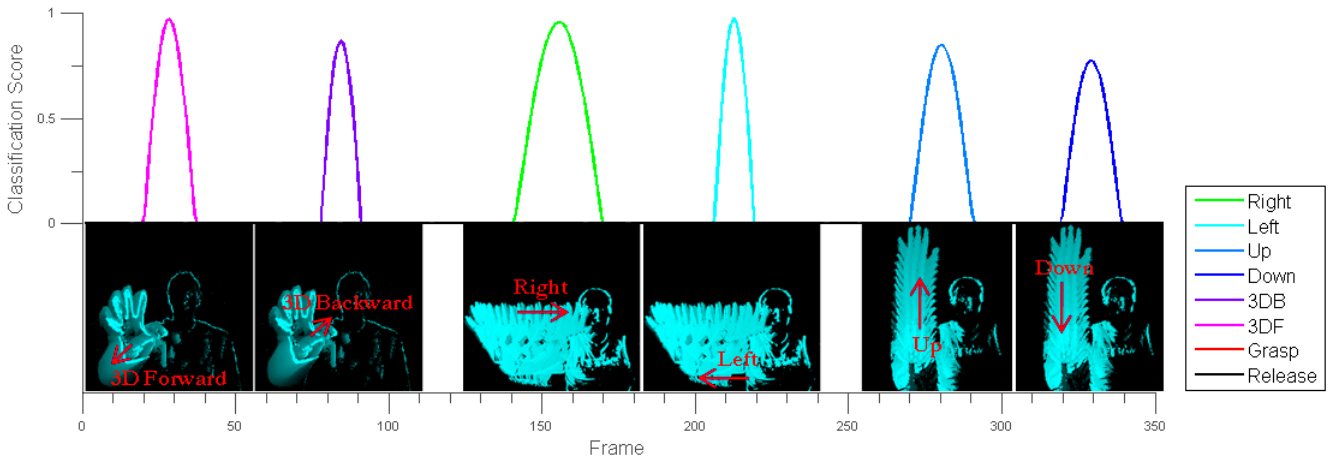

Fig. 3. The Figure illustrates the SVMs scores (Eq. 7) computed in real-time at each time step T over a sequence of 350 frames. A sequence of 6 primitives have been performed. Only positive classification scores are shown. The resulting sequence of primitives is $Forward$, $Backward$, $Right$, $Left$, Up, $Down$. The consequent string is thus $\langle f\ b\ r\ l\ u\ d \rangle$

training phase we obtain a set of N linear SVM classifiers $f_1(\bar{\mathbf{B}}), \ldots, f_N(\bar{\mathbf{B}})$, where N is the number of the chosen primitives. It has been experimentally shown [24] that sparse coding exploits the prior knowledge encoded in the data to obtain meaningful and effective representations, which can be separated with simple linear classifiers, and in fact we observed a similar behaviour for the specific case of the representation we propose. The main advantage of employing linear SVMs is the constant complexity in testing, which is essential for real-time applications, while retaining high classification accuracy.

2) Recognition of Primitive Sequences: given an arbitrary test video composed of L frames, we need to attribute a class A_s to each feature descriptor $\mathbf{B}_T(t)$, $t = 1, \ldots, L-T$, to identify the action primitives which are being shown in the video sequence. Because of the low speed of human movements, the same primitive persists for a certain number of frames, hence it is possible to monitor the scores of the classifiers only at specific break points revealing the changes from one action to another. More specifically, consider a buffer $\mathbf{B}_T(t)$ of feature vectors as defined in (4), where $T < L$ is the buffer length. The score $H_s(\mathbf{B}_T(t))$ of each primitive action A_s for the buffer $\mathbf{B}_T(t)$ is defined as:

$$H_s(\mathbf{B}_T(t)) = W \star f_s(\mathbf{B}_T(t)) \tag{7}$$

where the $\star$ is the convolution operator and W is a low-pass filter. Fig. 3 depicts an example of the evolution of primitives scores. The observed buffer $\mathbf{B}_T(t)$ is said to be an instance of the primitive A_s, if

$$\begin{aligned} &H_s(\mathbf{B}_T(t)) > H_q(\mathbf{B}_T(t)) \quad \forall A_q, A_q \neq A_s \\ &s.t. \quad H_s(\mathbf{B}_T(t)) > \tau \end{aligned} \tag{8}$$

where τ is a threshold estimated during training, according to the trend of each primitive. The score H_s is an unknown function and we are interested in its variation defined as:

$$\Delta H_s(\mathbf{B}_T(t)) = \frac{H_s(\mathbf{B}_T(t)) - H_s(\mathbf{B}_T(t-\Delta t))}{\Delta t} \tag{9}$$

As soon as one score increases it follows that the performed primitive must be the one related to that score, and as far as that score decreases while others are increasing, it follows that a change in direction is occurring. When the testing video has been completely analysed, the online recognition algorithm returns a sequence $\langle A_{s_1}, \ldots, A_{s_k} \rangle$ of primitive actions (see Fig. 3) synthesised in a string.

C. Complex Arm-Hand Action Modelling & Recognition

In our grammar-based high level architecture, each primitive represents a symbol of an alphabet, and each complex action is modelled as a language. Since it is unlikely that an action during human-robot interaction is not representable by a regular language we decided to model complex actions with grammars of the third type. Essentially, we assume that each action can be described by a regular language, which generates all the possible sequences specifying the action.
There is a vast literature about learning of regular languages from samples; Gold [26] proved that, in order to learn a regular language, both negative and positive samples are needed. Years later Denis [27] showed that it is possible to learn a regular language from simple positive examples, where "simple" is defined according to the Kolmogorov complexity [28]. Nevertheless, we aim to develop a fast learning procedure, which should not require more than one example to model a new language. Hence, rather than showing the new action several times so that the system is able to automatically build the structure of the FSM, we prefer to impose constraints on such structure, enabling the learning of the action through an online one-shot demonstration.
We can impose, without loss of generality, that an action can be represented by

- A finite sequence of symbols. The action "Move object" can be represented by the language $(g\ r\ e)$ where $g \equiv Grasp$, $r \equiv Right$ and $e \equiv Release$.
- An infinite repetition of the basic word. The action "Hello" can be represented by the language $(r\ l)^+$ or

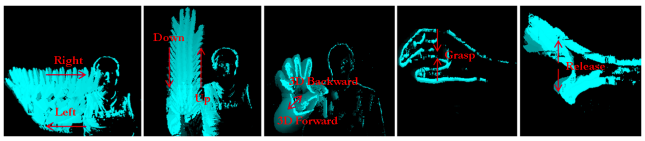

Fig. 4. The figure illustrates the chosen primitives in pairs: Right-Left, Up-Down, Forward-Backward, Grab and Release.

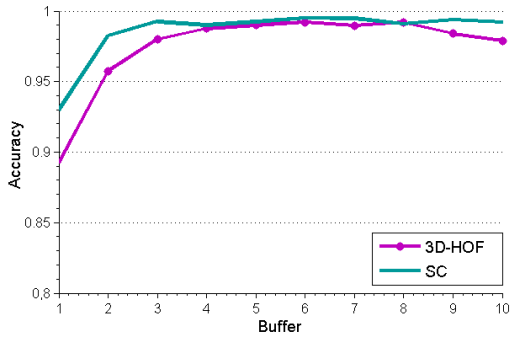

Fig. 5. The figure illustrates the accuracy results of the action primitives achieved wrt the length of the buffer for both 3D-HOF and SC descriptors. The bin and dictionary size are fixed $h = 4$ and $D = 256$ respectively. A standard K-fold cross-validation (with $K = 10$) procedure has been used for the evaluation.

$(r\, l\, ((r\, l)^* + (r\, l)^* l))$ where $l \equiv Left$, the operator s^* denotes all strings over the alphabet s, the operator s^+ is the subset of s^* containing all the elements of s^* except the empty string and the operator $+$ represents the union insiemistic operation.

- An union of different languages. The action "Square" can be represented by the language $(r\, u\, l\, d) + (r\, d\, l\, u)$ where $d \equiv Down$ and $u \equiv Up$.
- A composition of different languages. The action "Shake the hand" can be represented by $(f\, g) \cdot (u\, d)^+ \cdot (r\, b)$ where $f \equiv Forward$, $b \equiv Backward$ and the operator $\cdot$ represents the concatenation operation.

Imposing these constraints on the structure of the FSM we can teach a new action to the system with just one online demonstration. To recognise complex actions, we add a confidence measurement to decide with a certain reliance whether a string belongs to a language or not. Let s be the input string computed by the method described in Sec. II-B.2, and let n be its length. We generate all the possible strings belonging to the previously learned languages, with length at most equal to n. Let $s_1, \ldots, s_k$ be the set of the generated strings, we choose the one that satisfies:

$$\min_i \; L(s, s_i) \tag{10}$$

where L is the Levenshtein distance [13]. This metric allows retrieving a confidence measurement, so that the action with highest confidence is chosen, and those with a low confidence are discarded.

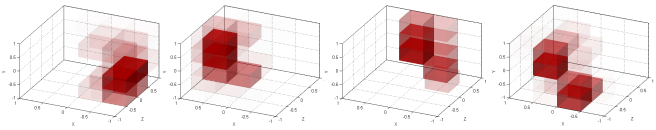

Fig. 6. The figure illustrates high level statistics (3D-HOFs) retrieved by the proposed scene flow algorithm. Starting from the left we show the histogram of the scene flow directions at time t, for the primitives: $Right$, $Left$, $Forward$, $Backward$ respectively. Each cuboid represents one bin of the histogram, for visualisation purposes we divided the 3D space in $n \times n \times n$ bins with $n = 4$. Filled cuboids represent high density areas.

III. Experiments

The system described so far has been implemented and tested on the iCub, a 53 degrees of freedom humanoid robot developed by the RobotCub Consortium [29]. It is equipped with force sensors and gyroscopes, and it resembles a 3-years old child. The iCub robot mounts two Dragonfly cameras, providing the basis for 3D vision, thus after an offline camera calibration procedure we can rely on a full stereo vision system. We define 8 primitives, as shown in Fig. 4. We impose the robot to stay still during the learning and recognition phases, and vice versa while it is acting the recognition is not performed.

A. Primitive Analysis

Since we are working with the iCub, we chose action primitives particularly suitable to describe manipulation tasks. We designed a dataset composed of 8 action primitives, as shown in Fig. 4. Data acquisition has been carried out recording subjects that performed 50 repetition for each action of the 8 possible primitives. Thus the dataset contains 400 examples as naturally executed by each human subject. We used a K-fold cross-validation procedure to evaluate the accuracy of the classifiers on isolated primitives. Fig. 5 shows that *for primitives* both 3D-HOF and SC descriptors achieve high classification accuracy at about 99%. We evaluated empirically that a bin size $h = 4$ (leading to a frame descriptor $\mathbf{z}(t) \in \mathbb{R}^{64}$) and dictionary size $D = 256$ are sufficient in order to achieve high accuracy. These good results were expected due to the high discriminative power of the 3D-HOFs (Fig. 6) on our primitives, which leads to a linearly separable set.

B. Recognition Algorithm Evaluation

Even though the action primitives are perfectly classified by our model, the performances of our online recognition algorithm should be evaluated considering a sequence of two or more actions. In fact a typical human action consists of a continuous sequence, which contains a certain amount of ambiguity across primitives. Therefore we assessed the accuracy of our algorithm on variable length and speed sequences of action primitives, demanding high performances in terms of both detection and classification. The proposed online recognition algorithm returns a sequence $S = \langle A_{s_1}, \ldots, A_{s_k} \rangle$ of primitive actions, that is modelled as a string. Given the ground truth $S^* = \langle A^*_{s_1}, \ldots, A^*_{s_k} \rangle$ of the performed

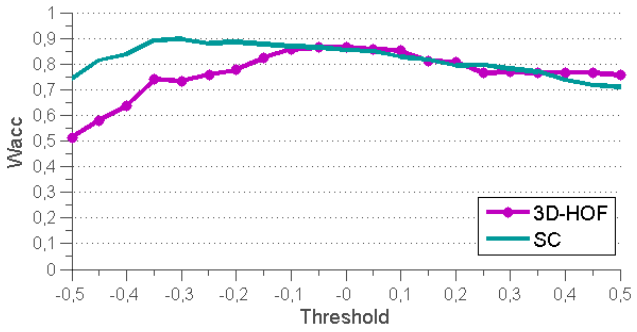

Fig. 7. The figure illustrates the accuracy results for sequences of action primitives (see Fig. 4) achieved with respect to the threshold parameter for both 3D-HOF and SC descriptors. We evaluated the performances on more than 100 actions composed of sequences of 1 to 6 actions.

sequence representing an instance of a regular language, the goal is to find a similarity function between the two strings. The problem reduces to distances between strings, hence a common metric for speech recognition systems reveals to be appropriate. We use the Word Error Rate (WER), based on the Levenshtein distance and defined as:

$$WER = \frac{S + D + I}{N} \tag{11}$$

where S is the number of substitutions (misclassifications), D the number of deletions (false negatives), I the number of insertions (false positives) and N the length of the ground truth string. In Fig. 7 we show the accuracy defined as $W_{acc} = 1 - WER$ according to the threshold parameter (see Eq. 8). This accuracy has been calculated on more than 100 actions composed of sequences of 1 to 6 actions. Notably, the maximum accuracy is around 90%, and it is obtained with a threshold chosen on the basis of SVMs' response.

C. Complex Action Analysis

We defined 10 complex actions, particularly suitable for our scenario, for instance "Move the object to the right", or even the simple infinite action "Hello"; we test them having 5 actors performing each complex action 10 times. The recognition procedure presented in Sec. II-C and based on the minimum Levenshtein distance between the test string and all the possible strings generated by the previously learned complex actions, meaningfully increases the accuracy. In fact, this confidence measurement brings the overall performances of our system to 96%.

IV. Discussion

The proposed system is a compositional based architecture for 3D arm-hand action modelling, learning and recognition. It achieves high performances running in real-time (∼15fps on 2.4Ghz Core 2 Duo Processor), and it performs quickly and reliably. The 3D-HOF descriptors enhanced with sparse coding techniques seem to be good candidates for 3D action modelling. The proposed syntactic approach enables not only the learning of an arbitrary number of complex actions, but above all one-shot learning of new actions. Moreover, the confidence measurement defined in Sec. II-C may provide an easy rejection scheme for words (i.e. actions) that do not belong to any learned language. We tested our system on the iCub robot, where it is meant to be incorporated into a more general human-robot interaction architecture providing active vision, attention, and gesture generation.

References

[1] C. Breazeal, "Robots that imitate humans," *Trends in Cognitive Sciences*, 2002.
[2] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys*, 2011.
[3] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, 2010.
[4] V. Krüger, D. Kragic, and C. Geib, "The meaning of action a review on action recognition and mapping," *Advanced Robotics*, 2007.
[5] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *TPAMI*, 2001.
[6] S. R. Fanello, I. Gori, and F. Pirri, "Arm-hand behaviours modelling: from attention to imitation," in *ISVC*, 2010.
[7] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," in *CVPR*, 2007.
[8] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.
[9] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *NIPS*, 2007.
[10] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model," in *CVPR*, 1992.
[11] P. Natarajan and R. Nevatia, "Coupled hidden semi markov models for activity recognition," in *Workshop Motion and Video Computing*, 2007.
[12] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *CVPRW*, 2010.
[13] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics - Doklady*, 1966.
[14] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *CVPR*, 1999.
[15] I. Laptev and T. Lindeberg, "Space-time interest points," in *ICCV*, 2003.
[16] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from a single depth image," in *CVPR*, 2011.
[17] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *TPAMI*, 2008.
[18] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, "Highly accurate optic flow computation with theoretically justified warping," *IJCV*, 2006.
[19] B. K. P. Horn and B. G. Shunk, "Determining optical flow," *Artificial Intelligence*, 1981.
[20] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic Scene Flow Computation for 3D Motion Understanding," *IJCV*, 2010.
[21] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *ICCV*, 2007.
[22] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *13th Scandinavian Conference on Image Analysis*, 2003.
[23] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, 2006.
[24] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009.
[25] V. Vapnik, *Statistical Learning Theory*. John Wiley and Sons, Inc., 1998.
[26] E. Gold, "Language identification in the limit," *Information and Control*, 1967.
[27] F. Denis, "Learning regular languages from simple positive examples," *Machine Learning*, 2001.
[28] A. N. Kolmogorov, "Three approaches to the quantitative definition of information." *Problems in Information Transmission*, 1965.
[29] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *WPMIS*, 2008.