4th International Workshop on Collaborative Robots and Human Robot Interaction (CR-HRI 2013)
International Conference on Collaboration Technologies and Systems (CTS 2013)
San Diego, USA, May 20-24, 2013

Robot Anticipation of Human Intentions through Continuous Gesture Recognition

Giovanni Saponaro
Institute for Systems and Robotics
Instituto Superior Técnico, UTL
Lisbon, Portugal
gsaponaro@isr.ist.utl.pt

Giampiero Salvi
KTH, Royal Institute of Technology
Speech, Music and Hearing
Stockholm, Sweden
giampi@kth.se

Alexandre Bernardino
Institute for Systems and Robotics
Instituto Superior Técnico, UTL
Lisbon, Portugal
alex@isr.ist.utl.pt

***Abstract*—In this paper, we propose a method to recognize human body movements and we combine it with the contextual knowledge of human-robot collaboration scenarios provided by an *object affordances* framework that associates actions with its effects and the objects involved in them. The aim is to equip humanoid robots with action prediction capabilities, allowing them to anticipate effects as soon as a human partner starts performing a physical action, thus enabling interactions between man and robot to be fast and natural.**

We consider simple actions that characterize a human-robot collaboration scenario with objects being manipulated on a table: inspired from automatic *speech* recognition techniques, we train a statistical *gesture* model in order to recognize those physical gestures in real time. Analogies and differences between the two domains are discussed, highlighting the requirements of an *automatic gesture recognizer* for robots in order to perform robustly and in real time.

I. Introduction and related work

In recent years, there has been a surge of interest in interfaces whereby users perform uninterrupted physical movements with their hands, body and fingers to interact with smartphones, game consoles, kiosks, desktop computer screens and more [1]. At the same time, the number of autonomous service robots integrated in society —as opposed to industrial ones— is ever-increasing. During 2011, 2.5M such robots were sold: 15% more than in the previous year.[a]

Given these premises, it is important to develop pattern analysis techniques suited for recognizing physical gestures in the context of task-based human-robot collaboration. This article presents a vision-based approach to classifying human task actions toward enabling robots to provide appropriate support to humans, as illustrated in Fig. 1, by using statistical models based on training data for recognizing real-time continuous gestures.[b]

This work is set within the object affordances framework [2], [3], which encodes causality relations between actions, objects and the effects of actions on objects. Our contribution to this probabilistic framework is that of explicitly modeling and measuring action variables, as shown in the upper part of Fig. 2. For a more detailed explanation of object affordances, see Sec. I-B.

[a]http://www.ifr.org/service-robots/statistics/

[b]In this article, the term "gesture" refers to intentional physical actions (see also Sec. I-A), and we will use it interchangeably with the term "action".

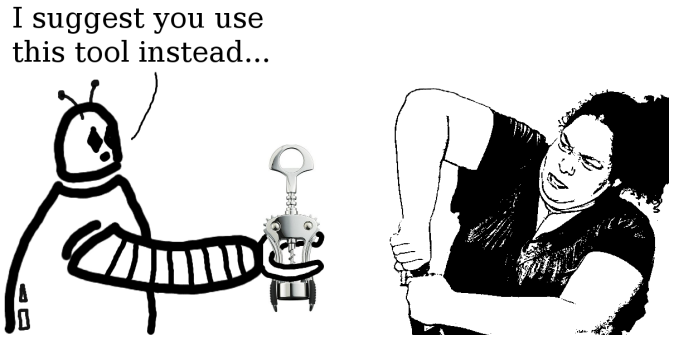


Figure 1: A robot capable of recognizing human gestures can intervene before the action is finished and provide help.

We argue that physical gestures give a hint of the intention over human action or aim, and we wish to capture this predictive power so that robots can exploit it for smoother interactions with their human counterparts. Our proposed system aims at equipping robots with the possibility of predicting human intentions by analyzing natural, continuous bodily gestures and the contextual knowledge expressed in object affordances, such as object shape and relationships between objects and movements with certain dynamics or trajectories. The whole system is sketched in Fig. 2, and in this paper we focus on the *continuous gesture recognition* aspect with analysis and results, on our planned experiments with a humanoid robot and on how gestures fit into the affordance network learned in previous work. Other components of the system, such as human-robot mimicking, will be discussed in future work.

In the remainder of this paper we outline the nature of dynamic gestures and related work in the automatic gesture recognition literature, we present our model and experimental results, and we show the feasibility of the proposed system in a possible human-robot interaction setting.

A. Automatic Gesture Recognition

In a broad sense, gesture is a component of human communication that involves the movements of different body parts: the whole body, hands, arms, fingers and/or face. It constitutes a primary modality for humans, who learn to use it as early as during their first year of age, before they learn

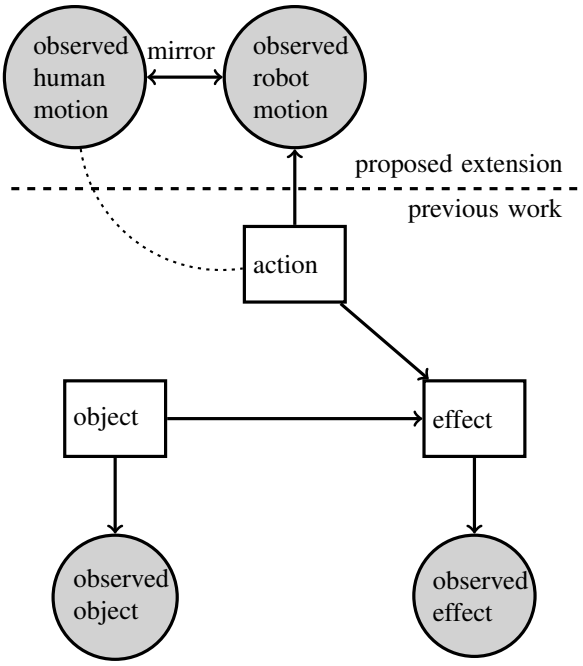


Figure 2: Affordance learning schema, with the proposed extension that takes human and robot movement into account. Because human actions and robot actions can be mirrored, the dotted line indicates a possible shortcut to connect human actions with action primitives directly. Square nodes are discrete-valued, round nodes are continuous, shaded nodes are observable through robot sensory data or computer vision, and edges indicate Bayesian dependency.

to speak [4]. The nature of human gestures is ambiguous and context-dependent [5]: there exist many-to-one mappings between gestures and conveyed concepts. In previous work [6], we studied human interpretation of robot gestures; in this work we tackle the opposite problem of how robots can recognize human gestures.

Different approaches have been proposed to design automatic gesture recognition systems, both to decide which *features* are salient for recognition [7] and which *model* best classifies them. For more comprehensive reviews of these systems, we refer the reader to [8]–[10].

Designing an automatic gesture recognizer poses two main issues:

1) spatio-temporal variability: the same physical gesture can differ in shape and duration, even for the same gesturer;
2) segmentation: the start and end points of a gesture are difficult to define and identify.

Common features for gesture recognition systems include: skin color segmentation, optical flow (the apparent visual motion caused by the relative motion of objects and viewer), arm-hand tracking in 2D or 3D, full body tracking.

Many gesture classifiers are designed to work in a controlled environment, or they make strong assumptions:

- limited and fixed lexicon of permitted gestures
- availability of the whole test data sequence to classify (system only works offline)
- constrained physical space (hands must move only within a certain region of upper body)
- unnatural interaction (isolated gestures, to be preceded and followed by a relaxed pose lasting several seconds)
- users must wear expensive hardware tracking devices.

Neuroscience experiments [11] have suggested that the area of the human brain responsible for gesture processing is also employed for speech processing, functioning in fact as a modality-independent semiotic system, connecting meaning to various types of symbols: words, gestures, images, sounds, or objects. In particular, we propose that the link between gesture and speech justifies the usage of tools that, as in automatic language recognition, (i) permit an abstraction hierarchy and (ii) are suited for capturing time series data. Hidden Markov Models, explained below, are one such statistical tool. We adopt an HMM-based approach to recognize human or robot gestures that follow *temporally dynamic patterns*.

Hidden Markov Models (HMMs) [12] are a statistical tool for modeling time series data. They have been applied to the segmentation and recognition of sequential data with spatial and temporal variability such as speech, machine translation, genomics and financial data. One of the advantages of HMMs —and a reason behind their popularity— is the fact that they are computationally tractable thanks to dynamic programming techniques: marginal probabilities and samples can be obtained from an HMM with the Forward–Backward algorithm, and the most likely sequence of hidden states can be estimated with the Viterbi algorithm.

A continuous-output HMM is defined by a set of states $\mathcal{S} = \{s_1, \ldots, s_Q\}$ and by a set of parameters $\lambda = \{A, B, \Pi\}$, where $A = \{a_{ij}\}$ is the transition probability matrix, a_{ij} is the transition probability from state s_i at time t to state s_j at time $t+1$, $B = \{f_i\}$ is the set of Q observation probability functions (one per state i) with continuous input and output, Π is the initial probability distribution for the states.

Selected related works in the dynamic gesture recognition literature are described in the remainder of this section.

The system by Yamato et al. [13] was among the first to apply HMMs for the recognition of human gestures and actions, using 25×25 pixel subsampled images of tennis strokes as features. However, this model required many ad-hoc pre-processing and filtering steps and the outputs were discrete, making the system not feasible or robust to be used in other domains. By contrast, our proposed approach does not assume prior filtering of the gestural feature points to reduce noise — in doing so, we preserve all the information contained in the raw data points, and we input these points to HMMs without pre-processing (noise is addressed by having enough diverse data samples of the considered scenarios); this way, we can execute the system in real time on a robot platform observing a *continuous* stream of human actions, not having to wait for the input gesture to be finished in order to recognize it.

Wilson et al. [14] developed a *parameterized* gesture rec-

ognizer where people's motion was recorded with a Polhemus motion capture system. This work is notable because it not only can detect a general pattern (e.g., human gait), but it is also able to extract context-dependent components of that pattern (e.g. speed, style). The main drawback lies in the high computational cost induced by the Parametric Hidden Markov Model and Generalized Expectation-Maximization algorithm during recognition. Our approach uses standard HMMs and EM, making it more feasible to be run online by a robot; in addition, the users of our human-robot interaction scenarios do not have to wear motion capture devices, but they can perform actions *naturally* in their everyday clothes.

Starner et al. [15] proposed a system to recognize *sign language*, where each sign word is associated to an HMM with an ad-hoc structure that fits their data (four states, each state can cycle to itself or proceed to the next one, and state 1 can jump to state 3 directly), features are determined with computer vision (users wear colored gloves), and a semantic grammar is used to check the validity of phrases. Our system does not require human users to wear colored attire or special hardware, and our HMMs can have a varying number of states with homogeneous transitions, as in Fig. 5b: not having skip transitions between particular state pairs permits us to modulate HMM parameters easily, handling gestural data of different nature, for example with different temporal durations. Alon et al. [16] also addressed the sign language recognition problem, using sophisticated visual motion features and a dynamic programming approach to prune multiple hypotheses, thus taking into account concurrent subgesture relationships;[c] their hand features are normalized with respect to the location and scale of the human's *face*, whereas our proposed approach centers them around the *torso* (human center of gravity), making it more versatile to recognize two gestures of the same class that occur at different horizontal distance from the face.

Another interesting work is the one by Lee and Kim [17], which analyzes gestures performed with one hand on a simple visual background, introducing the notion of a nongesture garbage threshold, similar to silence models in speech. However, their garbage model is ergodic (it is built by fully connecting all the states from all the gesture models in the system) and as such can incur in excessive computational burden, due to the high number of model parameters to optimize. Our proposed garbage model is more compact, being trained with a low number of states, just like another gesture model.

The article by Yang et al. [18] aims at recognizing complex actions (e.g. sitting on the floor, jumping) using angles between human body parts as features, then clustering them with Gaussian Mixture Models, *partitioning* the physical space into regions and then training HMMs for gestures and between-gesture transitions (garbage). One limitation of this approach is that the HMM states are tied to the pre-defined physical cluster or regions, thus this system cannot deal well with *scale* variations (e.g. two gestures conveying the same message, one with wide arms and the other one with narrow, less emphatic movements). Our approach is less sensitive to scale, because we train each gesture class with varying amplitude degrees and we let the model assign states to spatial points automatically, without clustering into regions.[d]

[c]For example, the "5" shape can also be interpreted as the first part of an "8" shape.

B. Object Affordances

The robot object affordances framework [2] takes its inspiration in psychology, and considers affordances as a mapping between actions, objects and the effects of actions on objects, as displayed in Fig. 2. Such mapping can be hand-coded, learned by demonstration, learned by robot self-exploration, or by a combination of these methods. Using inferential reasoning and Bayesian Networks (BNs, a probabilistic model that represents random variables and conditional dependencies on a graph), this framework allows robots to recognize objects or actions (intentions) given the observations of object features and motions, to predict the outcome of an action given the observation of an effect, or to plan actions in order to achieve a desired object/effect configuration.

Because the affordance network of Fig. 2 encodes the dependency relations between the variables, we can compute the marginal distribution of one or more variables, given the values of the other ones (it is not necessary to know the values of all the variables, in order to perform inference). An affordance network can be seen as a knowledge system which can be queried and contains three types of variables: A (actions), observed object features (F_i) and observed object effects (E_i). For example, to predict the resulting effects when observing an action a_i being performed on visual object features f_j, we have to compute $p(E|A = a_i, F = f_j)$.

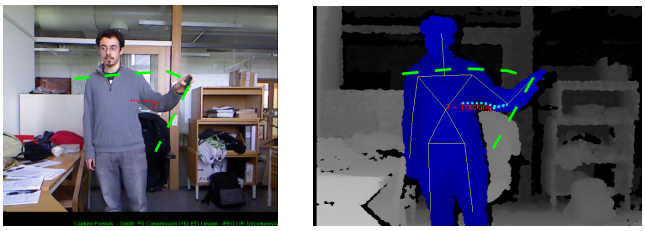
In [3], the robot affordance network of [2] was augmented with *spoken word* nodes, where each word may depend on any subset of A, F_i and E_i. This extension permits to associate words to meanings in robotic manipulation tasks in the event of co-occurrence between actions and verbal description of actions, object properties and resulting effects.

In [2] and [3], actions were not measured explicitly in the model: in fact, looking at the lower part of Fig. 2 (below the dashed line), the "action" node does not have any output edge that directly leads to an observation node. In this paper we present a possible way to extend robot affordances, improving action (intention) inference quality by the means of perceived gesture motion (above the dashed line). In previous work, the action (intention) could be inferred only indirectly through the observation of the effect – and after it had occurred. In the proposed approach, the action (intention) can be inferred before it completes, therefore giving the robot the opportunity to anticipate its effect and help the human accordingly. For

[d]We rely on gestural data to be diversified enough in amplitude (e.g. each movement having wide, narrow and medium-width examples), so that the trained Gaussian probabilities will cover the physical space reasonably well for the interaction scenarios that we consider. In the current version of our work, we do not claim real robustness to scale, which would require a feature space capable of interpolating between narrow and wide gestures (see Sec. II-A).

(a) Tap. (b) Grasp. (c) Push.

Figure 3: Some iCub robot gesture actions for object manipulation, with one hand reaching for an object for touching/grasping from different directions.



(a) RGB view. Hand trajectory shown in green, elbow trajectory in red.

(b) Depth skeletal view. Hand trajectory shown in green, elbow trajectory in light blue.

Figure 4: A dynamic human gesture, with temporal trajectory of selected joints being highlighted. The 3D coordinates of the joints of interest constitute the inputs of our statistical models of Fig. 5.

humanoid robots with morphology and motion capabilities similar to humans, we also consider the possibility to learn the gesture recognition model from movements of the robot itself. In that case, when observing the human, a map from human motion to robot motion must be taken into account, mimicking the role of mirror neurons [19] and allowing the robot to automatically imitate the human.

Some robot actions are shown in Fig 3, as performed by the iCub [20], a child-sized humanoid robot, possessing 53 degrees of freedom, facial expressions, tactile touch sensors, fully articulated eyes and head, and the ability to perform dexterous manipulation and gestures.

II. Proposed Approach

In this section we will formulate the action recognition model, its properties and training phase, and how to evaluate the presented tests. For the initial experiments presented hereafter, we assume:

- a human action *vocabulary* of three simple manipulation gestures (plus the "garbage" or "nongesture" action) that involve one arm, analogous to the ones of Fig. 3;
- a *feature space* containing the 3D position coordinates of the hand joint in time, obtained with an RGB-D camera;
- *inputs*: sequences of observation vectors as described in the previous points;
- *outputs* consisting of either (i) the recognized, most likely single action within an observation sequence or subsequence segment (Forward–Backward algorithm), or (ii) the estimated sequence of actions (Viterbi algorithm).

A. Feature Selection

The features we use to train our gesture classifier are computed directly from the spatial 3D coordinates of one or more human(oid) joints being tracked (hands, optionally also elbows, shoulders, torso, head), and they can be calculated online without having to wait for an input sequence to be finished. For this reason, we perform no normalization or filtering that requires knowledge of the completed sequence (e.g. global minima and maxima). The 3D joints coordinates can be obtained with general-pupose RGB-D cameras like the Microsoft Kinect or the Asus Xtion Pro, or with specialized computer vision algorithms. Fig. 4 illustrates the idea of extracting a time series of 3D coordinate features from a dynamic gesture.

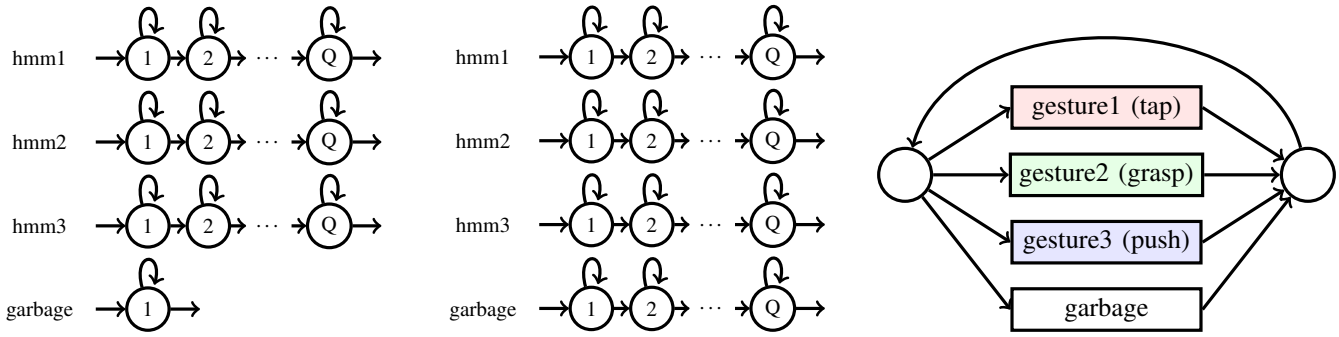
For the simple one-hand actions shown in Figs. 3 and 4, tracking one hand/arm is sufficient. While we do not apply normalization steps to the coordinates, we do apply a simple geometric transformation to the coordinates obtained with RGB-D cameras and skeleton recognition algorithms: we set our reference frame to be centered on the human *torso*, instead of the default sensor-centered reference frame. This transformation has two motivations, a theoretical and a practical one: from a theoretic perspective, it is coherent with the "human-in-the-loop" model, placing a virtual mobile point on the human user, and not on a fixed point attached to a camera or to a corner of an experiment room; from a practical perspective, this transformation provides *invariance to starting point* of a physical gesture. In other words, the user can perform actions at any distance or angle from the robot sensors, and these actions will always be measured with regards to his torso coordinate.

To disambiguate the gestural "words" of a domain, it is sometimes beneficial to enrich the feature space to include not only raw 3D coordinates of the joints of interest, but also their first and second derivatives [7], curvature, other structural geometric representations, and context-specific features (e.g. distance to interaction partner, distance to object to manipulate). In our current scenarios, however, we simply employ the 3D coordinates of the most meaningful joint (the hand), because it yields the highest recognition rate in initial tests.

B. Trained Models

We now present three different graphical models that were used in our experiments. The first two models serve as a baseline, while the third one is the final proposed approach, because it is powerful enough to capture a continuous (uninterrupted) sequence of actions, with permitted passages from one action to another being defined in its transition rules.

The first statistical model that we defined for our experiments ("Model 1") consisted of a Gaussian Mixture Model (GMM: a linear superposition of Gaussian components) –either trained with all the data, both gestural and nongestural, or trained with nongestural data only– and several HMMs,



(a) Model 1. (b) Model 2. (c) Model 3.

Figure 5: "Model 1": Hidden Markov Models trained with data from specific gestures, Gaussian Mixture Model trained with garbage (non-gesture) data.
"Model 2". Hidden Markov Models trained, respectively, with first gesture data, with second gesture data, with third gesture data and with nongesture (garbage) data. Each model is independent from the other ones, therefore it can have arbitrary state indexes $1, \ldots, Q$, with Q not necessarily the same number for all the models.
"Model 3". Hidden Markov Models (previously trained, respectively, with first gesture data, with second gesture data, with third gesture data and with nongesture/garbage data) after being merged. Each rectangle represents a gestural HMM like the ones shown in Fig. 5b, however in this case the states must be uniquely numbered.

each one trained for one gesture, as illustrated in Fig. 5a. This type of model allows to quickly test a gesture recognizer, clearly separating between the garbage part from the gesture part of a data sequence. On the other hand, the GMM nature of the garbage model does not allow to capture the dynamic nature which is also present in between-gesture transitions.

A second statistical model that we trained, "Model 2", improves on the previous model in the gesture/nongesture separation criterion. Here, the garbage model consisted of an HMM trained with garbage data, and other HMMs for actual gestures, as in Fig. 5b. In the current version of our work, for simplicity we have fixed the number of states Q to be equal for all gestures.

So far, we have considered the models of Fig. 5b to be independent from each other: each of them has its start, intermediate and final states, as well as its own prior probabilities, state transition probabilities and observation probabilities. In Fig. 5c, we have merged those models into one HMM with many states and appropriately combined probability matrices ("Model 3"). Merging the previously trained statistical models into one new HMM entails the following steps:

- weights matrix, means matrix, covariance matrix: concatenation of previous models' matrices along the Q dimension;
- initial probability vector: stochastic concatenation of previous models' priors, i.e., a column vector with $(Q \cdot \#\text{gestures})$ entries, all set to zero except for the first state of each gesture, set to $1/\#\text{gestures}$;
- transition matrix: $(Q \cdot \#\text{gestures}) \times (Q \cdot \#\text{gestures})$ block diagonal matrix built from the previous $(Q \times Q)$ matrices, allowing transitions from each of the previous HMMs' end states into the first state of any previous HMM (this allows the continuous gesture recognition algorithm to enter a sequence j at the end of any finished sequence i).

In all of the models described above, HMMs were trained with the incremental *mixture splitting* technique, inspired from speech recognition, in order to obtain the desired number of output Gaussians M_{des}. Initially the mixture has $M = 1$ Gaussian (with mean initialized to empirical mean and covariance initialized to empirical covariance of gesture data, respectively); we run the Baum–Welch algorithm[e] to improve HMM parameter estimates; then we enter a cycle, in which we run UPMIX (adapted from [21, Sec. 10.6], sketched in Alg. 1) and Baum–Welch, increasing the counter M; the cycle terminates when the weights matrix contains M_{des} Gaussians as desired. This technique allows us to achieve higher likelihoods than with simple Baum–Welch (EM), as shown in Fig. 6.

In the current version of our work, we collected training data of one person performing actions similar to the robot gestures depicted in Fig. 3 without the manipulated objects (because they are not considered at this stage), in other words we trained the action recognizer with action *pantomimes*. Each action was performed in three different amplitude classes: wide gestures (emphatic arm movements), medium-width gestures and narrow gestures (subtle movements). Each amplitude class was acquired multiple times (12–14 times), thus providing around 40 training repetitions for each of the manipulation actions considered. This data set was used to train all the statistical models described in this section.

In the next section, we show recognition results obtained by employing common HMM inference methods [12]: (i) Forward–Backward algorithm for isolated gesture recogni-

[e]The Baum–Welch algorithm is an instance of the Expectation–Maximization (EM) algorithm used to estimate HMM parameters.

Algorithm 1 Gaussian mixture splitting.

procedure UPMIX(weights, means, covariances)
    weights: split heaviest entry
    means: duplicate corresponding entry
    means: perturb new entries to be $\text{means}_{1,2}(i) \pm = \sqrt{\text{cov}(i,i)} \cdot \text{pertDepth}$ ▷ pertDepth = 0.2
    covariances: duplicate corresponding entry
    $\text{M} := \text{M} + 1$ ▷ M: current no. of Gaussians
end procedure

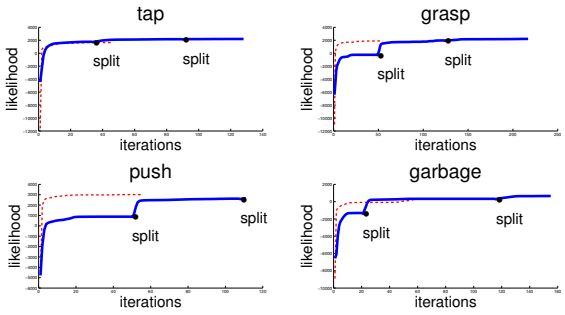


Figure 6: Evolution of the likelihoods of the models, comparing Expectation–Maximization algorithm when initialized with M=3 Gaussian outputs from the headstart (dashed red line) and when employing the mixture splitting technique (solid blue line, with points where the number of mixtures was incremented being highlighted). This affects the whole model, because the splitting applies to each GMM, but the retraining requires the full HMM. With the exception of the "push" gesture class, our method achieves a higher likelihood than simple EM.

tion, which computes the most likely single action recognized from a test data sequence; the major downside of this technique is that it requires the segmentation of test data, thus the availability of all test data offline; (ii) Viterbi algorithm for continuous gesture recognition: this method does not require prior segmentation of test data, and it outputs the estimated sequence of actions (state path) that best explain the test data sequence.

III. Experimental Results

Gesture recognition tests for the different models and algorithms are shown in Figs. 7 for the baseline Models 1 and 2, in Figs. 9 and 10 for the proposed approach which uses Model 3. Both training and test sequences were collected by the authors using an RGB-D camera recording gestures from one person. While we have yet to test how robust the system is to people with different heights and sizes, we expect it to be robust because we are applying a normalization step in all the observed measurements, dividing them by average shoulder width after a few frames (this can be done in real time). The feature space that we use in the current version of the work coincides with the 3D position coordinates of the hand joint in time; enriching the space with the coordinates of other joints such as shoulder and elbow actually decreased the recognition performance in our tests.

Forward–Backward classification results with "Model 1" are shown in Fig. 7a. The test sequence consists of nine continuous gestures, specifically three triplets (tap, grasp, push), the first triplet occurring at slow speed, the next one at medium speed, and the final one at fast speed. In this experiment, the test sequence was segmented similarly to how training data was segmented. In general, this is not safe to assume in a real time scenario, unless a delay is added. The problem here is that the gesture threshold is "too strict", voiding many HMM assignment classifications, even where they are correct.

In the "Model 1" experimental setup described above, gesture recognition performs poorly, with a recognition rate below 50%, mainly due to the fact that the garbage GMM cannot learn the temporal nature of nongesture (between-gesture) transitions.

Taking "Model 2" (Fig. 5b) into account, Fig. 7b displays improved Forward–Backward classification results. Compared to "Model 1", this model is better in correctly separating garbage segments from gesture ones, which we expected because the gesture classifier is richer here, being able to capture the dynamic nature of between-gesture transitions with its dedicated HMM. However, classification still suffers during probabilistic gesture class assignment, confusing taps with grasps for all velocities of the input sequence.

"Model 3" (Fig. 5c) allows us to illustrate the performance of our system with the Viterbi algorithm results of Figs. 9 and Fig. 10. The algorithm reconstructs the optimal (most likely) gesture state path resulting from a given test sequence. In these experiments, we assume that the context is described as the human-robot manipulation scenario shown in Fig. 8, whereby a user has to correctly move and grasp an object on a table, without making it collide with other objects: the correct strategy (intention) corresponds to the Push-Tap-Grasp sequence, a fact known *a priori* by the system. In Fig. 9 (left), the recognition accuracy is high (actions are detected in the correct temporal regions, and they are classified correctly 3/3 times) and the intention of the user is inferred to be coincident to the correct Push-Tap-Grasp strategy. On the other hand, Fig. 9 (right) shows a case where the recognition is still correct (the action sequence is correctly identified as Tap-Push-Grasp), but the wrong intention or strategy on the part of the user can be detected – thus allowing the robot to intervene, as motivated by the scope of this paper. Finally, Fig. 10 shows a

(a) Initial configuration.

(b) Intermediate configuration.

(c) Final configuration.

Figure 8: Example scenario to be applied in a human-robot collaboration setting: a human user sitting on the left has to move the mug next to the bottle, avoiding the red obstacle on the table, so that a robot bartender can fill the mug. The repertoire of permitted actions corresponds to the three gestures of Fig. 3. Without delving into the planning problem, which is out of the scope of this paper, we assume that the robot system knows that Push-Tap-Grasp is the correct strategy considering the initial table configuration, while for instance Tap-Push-Grasp is an incorrect strategy due to constraints. Fig. 9 (left) and Fig. 9 (right) reflect these two situations from the pattern recognition perspective.

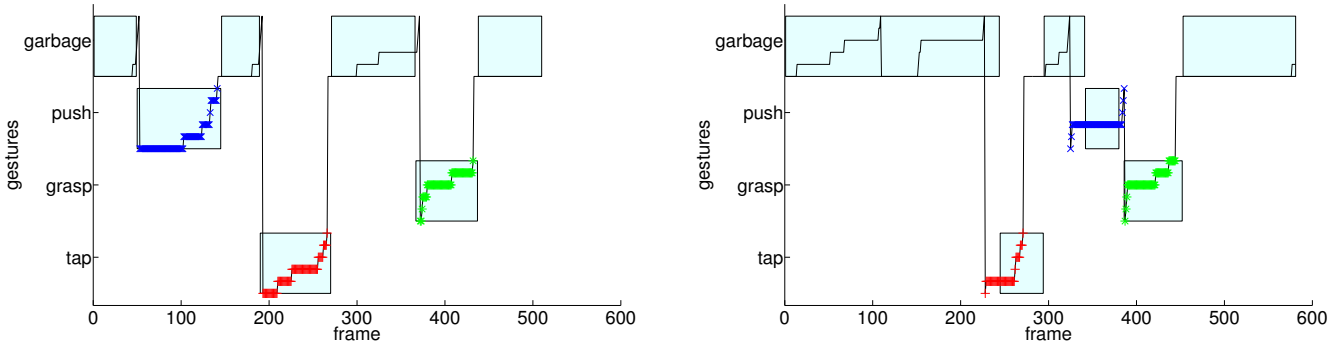


Figure 9: Two results of the proposed human-robot manipulation scenario of Fig. 8. Red plus signs: tap states, green stars: grasp states, blue crosses: push states, rectangles: human-labeled ground truth segmentation.
Left: a Push-Tap-Grasp action sequence performed by the user is correctly recognized (3/3 score), the user intention is found to be correct too, meaning that it is feasible given context and a table/object configuration. Right: a Tap-Push-Grasp action sequence is correctly recognized (3/3 score), although the user intention can be detected by the system as being incorrect considering the current context – allowing the system to alert the user.

test sequence which the system failed to recognize correctly as Push-Tap-Grasp (the order of actions actually performed by the user), due to limitations in training data, in the sensor we use and in the general statistical robustness of our current model.

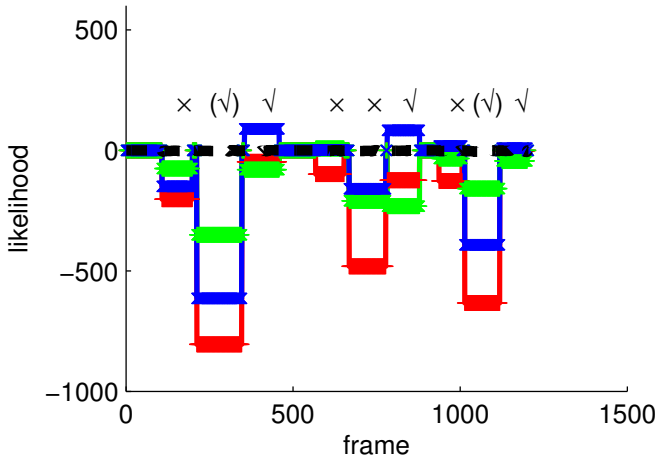
IV. Conclusions

Gestures are a paramount ingredient of communication, tightly linked with speech production in the brain: the ability to interpret the physical movements of others improves the understanding of their intentions and thus the efficiency of interactions. We propose a method to recognize gestures in a continuous, real time setting with statistical methods, and we discuss how to incorporate the predictive power supplied by human actions into robot affordance learning, ultimately allowing robots to anticipate others' intentions while interaction partners are still performing their actions.

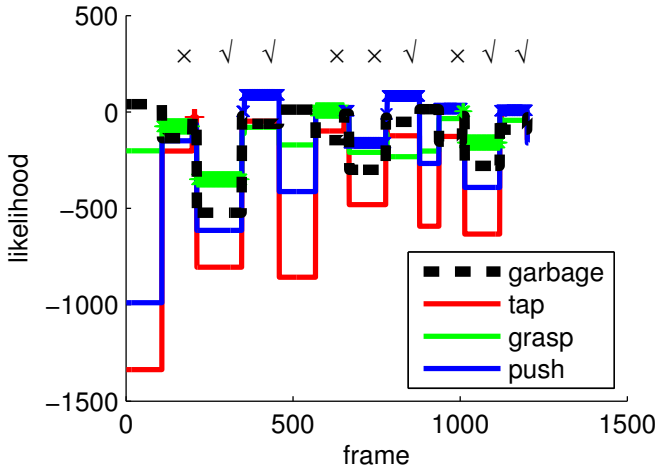
This article laid the foundations for adding action knowledge in interactive affordance scenarios. Different probabilistic models for gesture recognition were discussed and tested in an object manipulation scenario, with encouraging results. Future work includes performing tests in human-robot object manipulation tasks, enriching the actions repertoire with more complex gestures taken from contexts different than object manipulation (e.g. kitchen activities), and mirroring human and robot actions through optical flow methods.

Acknowledgment

Work partially supported by the Portuguese Government – Fundação para a Ciência e a Tecnologia (PEst-OE/EEI/LA0009/2011) and European Commission project POETICON++ (FP7-ICT-288382). G. Saponaro is supported by an FCT doctoral grant (SFRH/BD/61910/2009). Fig. 1 was elaborated with permission from Flickr user john00879.



(a) Model 1 (Fig. 5a) performance on segmented input sequence.



(b) Model 2 (Fig. 5b) performance on segmented input sequence.

Figure 7: Likelihood computed with Forward–Backward algorithm. $\surd$: correct gesture classification, $\times$: wrong classification, $(\surd)$: classification is correct but is voided by GMM nongesture threshold.

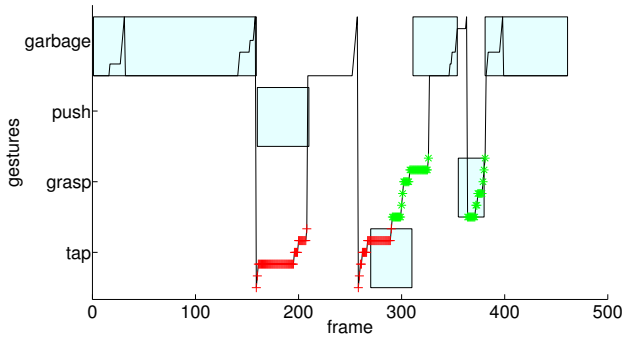


Figure 10: An outcome of the human-robot manipulation scenario showing the limitations of our approach in the presence of noise: the Push-Tap-Grasp action sequence performed by the user is not correctly classified by the statistical model. Red plus signs: tap states, green stars: grasp states, blue crosses: push states, rectangles: human-labeled ground truth segmentation.

References

[1] D. Wigdor and D. Wixon, *Brave NUI World: Designing Natural User Interfaces for Touch and Gestures*. Morgan Kaufmann, 2011.

[2] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning Object Affordances: From Sensory–Motor Coordination to Imitation," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, Feb. 1998.

[3] G. Salvi, L. Montesano, A. Bernardino, and J. Santos-Victor, "Language Bootstrapping: Learning Word Meanings From Perception–Action Association," *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 3, pp. 660–671, June 2012.

[4] M. Tomasello, M. Carpenter, and U. Liszkowski, "A New Look at Infant Pointing," *Child Development*, vol. 78, pp. 705–722, 2007.

[5] D. McNeill, *Hand and Mind: What Gestures Reveal About Thought*. University of Chicago Press, 1992.

[6] G. Saponaro and A. Bernardino, "Generation of Meaningful Robot Expressions with Active Learning," in *ACM/IEEE Int'l Conference on Human-Robot Interaction (HRI)*, Lausanne, Switzerland, March 2011.

[7] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick, and A. Pentland, "Invariant Features for 3-D Gesture Recognition," in *IEEE Conference on Automatic Face and Gesture Recognition (FG)*, 1996, pp. 157–162.

[8] Y. Wu and T. S. Huang, "Vision-Based Gesture Recognition: A Review," in *Int'l Gesture Workshop (GW)*, 1999, pp. 103–115.

[9] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," *IEEE Trans. Syst., Man, Cybern.*, vol. 37, no. 3, pp. 311–324, 2007.

[10] C. Keskin, O. Aran, and L. Akarun, *Hand Gesture Analysis*. Springer-Verlag, 2011, ch. 6.

[11] J. Xu, P. J. Gannon, K. Emmorey, J. F. Smith, and A. R. Braun, "Symbolic Gestures and Spoken Language Are Processed by a Common Neural System," *Proceedings of the National Academy of Sciences of the USA*, vol. 106, no. 49, pp. 20 664–20 669, 2009.

[12] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.

[13] J. Yamato, J. Ohya, and K. Ishii, "Recognizing Human Action in Time-Sequential Images using Hidden Markov Model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1992, pp. 379–385.

[14] A. D. Wilson and A. F. Bobick, "Parametric Hidden Markov Models for Gesture Recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 884–900, 1999.

[15] T. Starner, J. Weaver, and A. Pentland, "Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 1371–1375, 1998.

[16] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "A Unified Framework for Gesture Recognition and Spatiotemporal Gesture Segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 31, no. 9, pp. 1685–1699, Sept. 2009.

[17] H.-K. Lee and J. H. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 10, pp. 961–973, Oct. 1999.

[18] H.-D. Yang, A.-Y. Park, and S.-W. Lee, "Gesture Spotting and Recognition for Human–Robot Interaction," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 256–270, 2007.

[19] M. Lopes and J. Santos-Victor, "A Developmental Roadmap for Learning by Imitation in Robots," *IEEE Trans. Syst., Man, Cybern. B*, vol. 37, no. 2, pp. 308–321, Apr. 2007.

[20] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano, "The *iCub* Humanoid Robot: An Open-Systems Platform for Research in Cognitive Development," *Neural Networks*, vol. 23, no. 8–9, pp. 1125–1134, 2010.

[21] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.4)*. Cambridge University Engineering Department, 2006.