2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Continuous Gesture Recognition for Flexible Human-Robot Interaction*

Salvatore Iengo[1] Silvia Rossi[1] Mariacarla Staffa[1] Alberto Finzi[1]

Abstract— **In this work, we present a reliable and continuous gesture recognition method that supports a natural and flexible interaction between the human and the robot. The aim is to provide a system that can be trained online with few samples and can cope with intra user variability during the gesture execution. The proposed approach relies on the generation of an ad-hoc Hidden Markov Model (HMM) for each gesture exploiting a direct estimation of the parameters. Each model represents the best prototype candidate from the associated gesture training set. The generated models are then employed within a continuous recognition process that provides the probability of each gesture at each step. The proposed method is evaluated in two case studies: a hand-performed letters recognizer and a natural gesture recognizer. Finally, we show the overall system at work in a simple human-robot interaction scenario.**

I. INTRODUCTION

In this paper, we present a novel approach to real-time and continuous gesture recognition that allows a flexible, natural, and robust human-robot interaction (HRI). The proposed system should support the social interaction between the human and the robot by enabling a continuous process of evaluation and interpretation of the reciprocal movements. Furthermore, the proposed methodology should permit an incremental development of the HRI system through simple training and modular insertion of new gestures.

In literature, we find several approaches to gesture recognition. Most of the them are based on statistical modelling, such as Principal Component Analysis (PCA), multi-dimensional Hidden Markov Models (HMM) [9], [11], [10], [12], Kalman filtering, and condensations algorithms. On the other side, Finite State Machines (FSM) has been effectively used in modeling human gestures [2], [1]. Connectionist approaches involving neural networks have been also explored, such as time-delay neural network (TDNN) [13]. In HMM approaches the models are employed to represent the gestures and their parameters are learned from the training data. Based on the most likely performance criterion the gestures can be recognized through evaluating the trained HMMs [11],[10] and [12]. FSM methods for gesture recognition have been proposed [2]. As reported in [1], following this approach, the structure of the model is first manually decided based on the observation of the spatial topology of the data. The model is then iteratively refined in two stages: data segmentation and model training. The recognition phase is typically accomplished using some string matching algorithm like the Knuth-Morris-Pratt [5]. As for the connectionist approaches, in [13] a time-delay neural network (TDNN) for continuous gesture recognition is used. TDNN is a multi-layer feedforward network that uses delays between all layers to represent temporal relationships between events in time. TDNN is learned in order to recognize motion patterns because gesture are spatio-temporal sequences of feature vectors defined along motion trajectories. All the methods described above have advantages and disadvantages: HMMs require the data to be temporally well aligned during the recognition phase, hence the problem of the gesture delimiter arises, TDNNs address the latter problem by exploiting temporal dependencies among the sequences, but the number of the involved parameters is typically high, while FSMs need a manual modelling of the pattern (e.g., a grammar). In addition, the connectionist approaches require a very large training set to train the corresponding gesture models (e.g., using gradient descent algorithm). In contrast with these methods, we focus on a novel method capable of quickly generalize a gesture model starting from a very small training set and perform continuous gesture recognition with a very high accuracy. This method integrates different techniques: clustering algorithm for gesture quantization, Levenshtein distance for gesture prototype election, and Hidden Markov Model for continuous gesture recognition. Ad-hoc Hidden Markov Models are then generated for each gesture exploiting a direct estimation of the parameters. Each model represents the best candidate prototype from the associated gesture training set. The generated models are then employed within a continuous recognition process that provides the probability of each gesture at each step. In order to assess the proposed system we tested it considering two benchmarks: a hand-performed letters recognizer and a natural gesture recognizer. The collected empirical results show the potential of the approach with respect to other methodologies in literature. Finally, we show the proposed recognition system at work in a typical human-robot interaction scenario.

II. SYSTEM OVERVIEW

In this section, we detail the gesture recognition process. It consists of two phases: 1) a training phase, where the user shows few samples of a given set of gestures, and 2) a recognition phase, where the system recognizes the gesture performed by the user. The gesture acquisition process consists of the following steps (see Figure 1): Data acquisition (from Kinect device at the sampling period of 100ms); Noise filtering (with a Monte Carlo particle filter

*The research leading to these results has received funding from the European Communitys FP7-ICT 287513 SAPHARI and FP7-ICT 600958 SHERPA.

[1]Università degli Studi Napoli Federico II Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione (DIETI) via Claudio 21, I-80125 Napoli, Italy - {salvatore.iengo,silvia.rossi,mariacarla.staffa,alberto.finzi}@unina.it

estimator); Feature vector extraction; Vector quantization with K-means clustering; Hidden Markov Model (HMM) parameters generation, and HMM evaluation for gestures recognition.

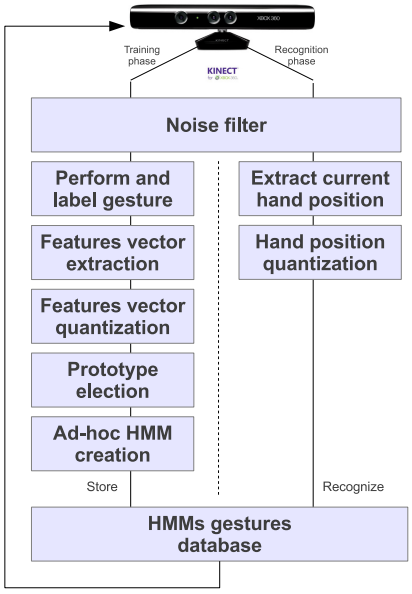


Fig. 1. System architecture.

A. Gesture definition.

We start defining the gesture dataset. Suppose that our gesture vocabulary consists of t gestures classes in the T gestures dataset. Our dataset is $\mathbf{T} = \{T_1, T_2, ..., T_t\}$ where every gesture class set T_j contains n_j instances so that $T_j = \{G_{1_j}, G_{2_j}, ..., G_{n_j}\}$ and n_j denotes the number of repetitions for each gesture of the given class j. Each gesture G_{c_j} is defined as $G_{c_j} = \{(x_{1_{c_j}}, y_{1_{c_j}}), (x_{2_{c_j}}, y_{2_{c_j}}), ..., (x_{m_{c_j}}, y_{m_{c_j}})\}$ where m_{c_j} denotes the number of coordinates belonging to the center of mass of the hand trajectory for the c-th repetition of the gesture belonging to the class j and $(x_{k_{c_j}}, y_{k_{c_j}})$ represents the k-th coordinate for the c-th gesture repetition for the class j.

B. Joints position estimation.

Due to the noise of the perceptive system (Kinect), the hand coordinates need to be smoothed over the time for each gesture. For this purpose (and to make the tracking system robust to occlusions) we deploy an importance sampling algorithm (see Algorithm 1). The state of the hand position at the current time-step k is obtained from the initial state and all the collected measurements $Z^k = \{z_i, i = 1..k\}$ once we solve the Bayesian filtering problem. That is, we need the posterior density $p(\mathbf{x}_k|Z^k)$ of the current state conditioned on all the measurements. As usual, the computation of $p(\mathbf{x}_k|Z^k)$ requires the definition of two phases associated respectively with prediction and update. In the first phase (prediction), we evaluate $p(\mathbf{x}_k|Z^{k-1})$, where the control $\mathbf{u}_k$ vector is defined as $\mathbf{u}_k = [v_{x_k}, v_{y_k}] = [\dot{x}_k, \dot{y}_k] = [\frac{\partial x_k}{\partial k}, \frac{\partial y_k}{\partial k}]$ is the velocity model (speed vector) of the hand in terms of speed among the two axes computed as the numerical derivative of two successive spatial position. In the second phase (update) we use a *measurement model* to incorporate information from the sensor to obtain the posterior probability density function $p(\mathbf{x}_k|Z^k)$ under the assumption of conditional independence of earlier measurements Z^{k-1} given x_k. The measurement model is given in terms of a likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ of the hand to be at location $\mathbf{x}_k$ given that $\mathbf{z}_k$ was observed. The posterior density $p(\mathbf{x}_k|Z^k)$ over $\mathbf{x}_k$ is obtained using the Bayes' Theorem.

Algorithm 1 OcclusionsAndNoiseFiltering(χ_{k-1}, $\mathbf{u_k}$, $\mathbf{z_k}$)

1: $\bar{\chi}_t \leftarrow \chi_t \leftarrow \emptyset$
2: **for** $m \leftarrow 1$ **to** M **do**
3: $\quad \mathbf{x}_k^{[m]} \leftarrow \mathcal{N}(\mathbf{x}_{k-1}^{[m]}, \alpha_0 \|\mathbf{u}_k\|^2) + \beta \mathbf{u}_k$
4: $\quad w_k^{[m]} \leftarrow \eta_0 (\|\mathbf{x}_k^{[m]} - \mathbf{z}_k^{[m]}\|)^{-1}$
5: $\quad \bar{\chi}_t \leftarrow \bar{\chi}_k + \langle \mathbf{x}_k^{[m]}, w_k^{[m]} \rangle$
6: **end for**
7: $m \leftarrow 1$
8: **while** $m < M$ **do**
9: $\quad q \leftarrow \eta_1 w_k^{[m]} M$
10: $\quad$ **for** $j \leftarrow 1$ **to** q **do**
11: $\quad\quad \mathbf{x}_k^{[m]} \leftarrow \mathcal{N}(\mathbf{x}_k^{[m]}, \alpha_1 (1 - w_k^{[m]})^2)$
12: $\quad\quad \chi_k \leftarrow \chi_k \cup \{\mathbf{x}_k^{[m]}\}$
13: $\quad\quad m \leftarrow m + 1$
14: $\quad$ **end for**
15: **end while**
16: **return** χ_k

Given $\mathbf{x}_k^{[m]}$ and $w_k^{[m]}$ computed as in the Algorithm 1, we get the approximation of the Equation (1) through the expected value of the distribution as reported in Algorithm 1. Algorithm 1 implements a Monte Carlo particle filter for position estimation. From the line 2 to 6 new particles with the associated importance weight are generated using the control and measurement vectors $\mathbf{u}_k$ and $\mathbf{z}_k$. The lines 8-14 update the particle using the importance weight previously computed.

$$\hat{\mathbf{x}_k} = E[p(\mathbf{x_k}|z_k, u_k)] \approx \sum_{m=1}^{M} \mathbf{x}_k^{[m]} w_k^{[m]} \tag{1}$$

Assuming that $\hat{\mathbf{x}}_{k_{c_j}} = \hat{x}_{k_{c_j}} \hat{y}_{k_{c_j}}$ is the corresponding estimation of the k-th hand position coordinate for the c-th gesture repetition for the class j computed by the Equation (1) we get the approximation robust with respect to raft and occlusions $G_{c_j} = \{(\hat{x}_{1_{c_j}}, \hat{y}_{1_{c_j}}), (\hat{x}_{2_{c_j}}, \hat{y}_{2_{c_j}}), ..., (\hat{x}_{m_{c_j}}, \hat{y}_{m_{c_j}})\}$.

C. Gesture quantization.

For the sake of simplicity, we temporarily replace the notation $G_{c_j} = \{(x_{1_{c_j}}, y_{1_{c_j}}), (x_{2_{c_j}}, y_{2_{c_j}}), ..., (x_{m_{c_j}}, y_{m_{c_j}})\}$ with $G = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$ and assume $\mathbf{x}_i = [x_i\, y_i], i \in [1, m]$. In this way, a gesture instance G is quantized as $Q(G) = \{q_1, q_2, ..., q_m\} : q_i = \arg\min_{k \in [1,K]} \|\mathbf{x}_i - \mathbf{C}_k\|, i \in [1, m]$, where $q_i \in [1, K]$, $C = \{(C_{1_x}, C_{1_y}), (C_{2_x}, C_{2_y}), ...,$

$(C_{K_x}, C_{K_y})\}$ is the set of the K centroids generated with a K-means algorithm over the whole dataset G (argmin defined with respect to the euclidean distance), and $\mathbf{C}_i = [C_{i_x} C_{i_y}]$, $i \in [1,K]$.

D. Generalized mean distance.

For each class of the original dataset, we define the distance d of two strings on the alphabet $\mathscr{A} = \{1,2,...,K\}$, where d is defined as $d : \mathscr{A}^* \times \mathscr{A}^* \to \mathbb{R}$ [1]. Suppose $A = \{s_1, s_2, ...s_n\} : s_i \in \mathscr{A}^*, i \in [1,n]$ is a set of n strings defined on the alphabet $\mathscr{A}$, the problem of finding the string with minimal distance from all the others is known as the Generalized Mean distance String (GMS). The distance metric we use is the Levenshtein distance metric algorithm reported in (2),

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+[a_i \neq b_j] \end{cases} & \text{else} \end{cases} \quad (2)$$

where, a and b are sequences and i and j are their indexes. Regardless the particular distance chosen, in [8] the authors demonstrated that the problem of finding the GMS is NP-Hard under Levenshtein distance for bounded and even binary alphabets. A more reasonable solution is to find the string, belonging to the set, that minimizes the sum of the distances above all the strings of a given class. This string is known as Set Median String.

$$SMS_j = \arg\min_a \sum_{i=1}^{n_j} lev_{a,Q(G_{i_j})}(|a|, m_{i_j}) : a \in Q(T_j) \quad (3)$$

E. Hidden Markov Model.

In this section, we describe how the HMM model of the gestures is generated.

A hidden Markov model (HMM) is a five-tuple (S, Σ, A, B, π), where $\{S\}$ is a set of states including the initial state S_1 and a final state S_F, Σ is the alphabet of the observation symbols, A is the transition probability matrix, $A = \{a_{i,j}\}$, $a_{i,j}$ is the transition probability from state i to state j, B is the output probability matrix, $B = \{b_j(O_k)\}$ (O_k stands for a discrete observation symbol) and π is the starting probability for each state.

Let $\lambda = (A, B, \pi)$ denote the parameters for a given HMM with fixed S and Σ, the key idea of our HMM-based gesture recognition is to use multi-dimensional HMM representing each defined gestures class c as a λ_c HMM model.

Therefore, a gesture is described by a set of N distinct hidden states and r-dimensional K distinct observable symbols.

The number of states of the HMM of the j-th gesture class is chosen to be equal to the SMS_j length as defined in 3.

[1] $\mathscr{A}^*$ denotes the sets of strings with zero or more repetitions of the elements belonging to the set A (Kleene operator)

Assuming $SMS_c = Q(G_{c_i}) = \{q_{1_{c_i}}, q_{2_{c_i}}, ..., q_{m_{c_i}}\}$, we have the following HMM model $\lambda_c = (A^c, B^c, \pi^c)$ for the c gesture class:

$$a^c_{i,j} = \begin{cases} p_{trans} & \text{if j=i+1} \\ 1 - p_{trans} & \text{else} \end{cases}, i \in [1, m_{c_i}], j \in [1, m_{c_i}] \quad (4)$$

$$B^c = \{b^c_j(o_k)\}, j \in [1, m_{c_i}], o_k \in [1,K] \quad (5)$$

$$b^c_j(o_k) = \begin{cases} p_{emit} & \text{if } o_k = q_{j_{c_i}} \\ \frac{1-p_{emit}}{K-1} & \text{else} \end{cases} \quad (6)$$

where A^c is the m by m matrix of the transition probabilities, π^c is the starting transition probability, S^c are the model states, p_{trans} and p_{emit} are, respectively, the transition and emit probability of the HMM.

F. Gesture recognition.

Given an observation sequence [2] $Q(G_{obs}) = \{q_{1_{obs_i}}, q_{2_{obs_i}}, ..., q_{m_{obs_i}}\}$ the best class is determined by $C(Q(G_{obs})) = \arg\max_c \; p(\lambda_c | Q(G_{obs}))$. We compute $P(\lambda_c | Q(G_{obs}))$ by applying the Bayes Theorem. Assuming $P(\lambda_c)$ and $P(Q(G_{obs}))$ constant, we can compute: $P(\lambda_c | Q(G_{obs})) \propto P(Q(G_{obs}) | \lambda_c)$. $P(Q(G_{obs}) | \lambda_c)$ can be computed through the Forward-Backward algorithm.

G. Continuous gesture recognition.

Continuous gesture recognition is much more complex than isolated gesture recognition, this is due to the difficulty in detecting boundaries among different gestures [14]. Here, we use a temporal sliding method as illustrated in Figure 2. For each new observation symbol the most likely belonging class is estimated. The algorithm is reported in 2, where for each class c the observation probabilities are computed through the Viterbi algorithm and the best match is returned.

Algorithm 2 ContinuousRecognition($obs_{1...n}$)

1: **for** $i \leftarrow 1$ **to** C **do**
2: Let $\mathbf{q}$ be the sequence of the last $||S_i||$ observation symbols where S_i is the set of the states of λ_i.
3: $p_i \leftarrow P(\mathbf{q} | \lambda_i)$
4: **end for**
5: **return** $argmax_i p_i$, $i \in [1,C]$

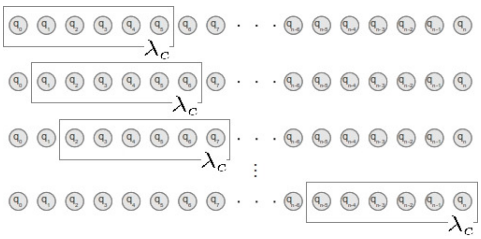


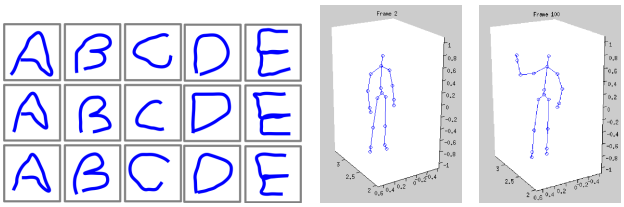
Fig. 2. An example of a temporal sliding for the gesture match.

[2] For simplicity we denote sequences as sets with implicit indexing for each element. It will be clear from the context if we are referring to sets or sequences.

III. CASE STUDY

In order to assess the system performance, two standard case studies have been considered: a letter recognizer and a natural gesture recognizer. The former is based on a subset of the English alphabet (A,B,C,D,E) while the latter is based on the Microsoft Research Cambridge-12 (MSRC-12) Gesture Dataset. Furthermore, in a final case study we illustrate the system at work in a HRI scenario.

Letter case study: In the letter case study, we aim at validating the intra user variability robustness. We choose the first 5 upper case letter of the alphabet: A,B,C,D and E. The Figure 3(a) shows how the user hand trajectory describes the five letters. During the training phase the user performed only 3 samples of the 5 letters. During the recognition process the user is asked to freely move and to perform 20 continuous gestures per each of the 5 letters (for a total of 100 gestures). In this setting we assume that a gesture c is successfully recognized if the likelihood of the c-HMM letter model overcomes a given threshold (set to 65% after empirical testing), it is rejected otherwise. In the training phase the gestures are showed and labeled to the system (supervised training) while in the recognition phase no explicit segmentation is required and the recognition takes place in a continuous gesture stream.



(a) Letters Trajectories (b) Frame 2 (c) Frame 100

Fig. 3. Examples of letter trajectories performed by the user hand and Frames of human skeleton.

MSRC-12 case study: The MSRC-12 gesture dataset consists of sequences of human skeletal body part movements (represented as body part locations) and the associated tags that should be recognized by the system. The dataset comprises 594 sequences, 719359 frames collected from 30 people performing 12 gestures. In total, there are 6244 gesture instances [4]. The gestures can be categorized into two abstract categories: iconic gestures - those that imbue a correspondence between the gesture and the reference (e.g. G2 - Crouch or hide (duck)), G6 - Shoot a pistol), and metaphoric gestures - those that represent an abstract concept (e.g. G1 - Start Music/Raise Volume, G3 - Navigate to next menu). The Figure 3(b)(c) shows a body skeleton taken at two successive time instants. In the experiment the 20% of the dataset was used as training set and the remaining 80% was used as test set. In particular, for each person performing a gesture the data set contains about 10 repetitions: the first 2 or 3 were used to train the model and the other 7 or 8 were used to test the results. The ground truth data is contained in separate files of the data set package - for each gesture performed there is a time stamp and a label for it. The evaluation was performed comparing the time stamp contained in the ground truth files and the time stamp provided by the proposed classification algorithm. A gesture is considered recognized if the label is the same of the ground truth and if the time stamp difference is not greater than 1 second.

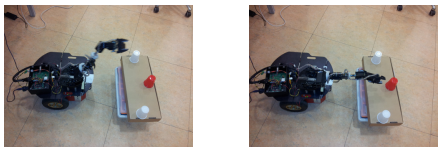


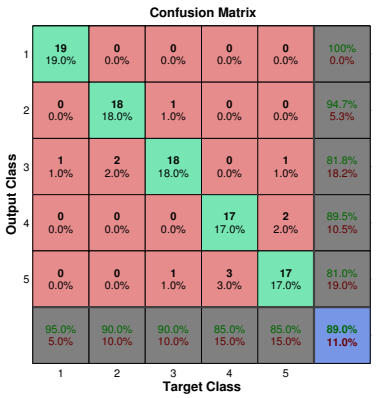
Fig. 4. Human-Robot Interaction case study: interaction task.

Human-Robot interaction case study: In this case study, we introduce a HRI setting where the task of the robot is to interpret and to execute the intentions of the human using only gestures. We considered the following simple task: a robotic arm is posed in front of a set of objects and is to decide which one to reach (see Figure 4); while executing the task the robot continuously monitors the human gestures to understand whether the current operative state is adherent with the human intention or not.

Initially, the robot slowly scans the possible targets moving the end-effector in different directions waiting for some stimulus from the human, who encourages the robot to move towards one of the targets. Once the manipulator starts to move towards one of the targets, depending on the recognized gesture probability, the robot can hesitate or move with confidence in the direction of the selected object. In this context, the robotic arm speed should depend on the confidence of the recognized gesture. When the human interaction starts to become uncertain or something unexpected happens, the robot can decide to stop the motion and switch towards another target. For the task three gestures are considered: *"GO ON"*, *"SLOW DOWN"* and *"SWITCH"*. The *"GO ON"* gesture is intended to suggest the robot to keep going on the current target (rotating the right arm in circle), the *"SLOW DOWN"* gesture makes the robot to decrease the approaching speed (moving right hand up and down) and *"SWITCH"* causes the robot to switch to the next target (moving right hand to left and then to right).

A. EXPERIMENTAL RESULTS

Letter case study results: In the Figure 5(a) the confusion matrix of the letters recognition task is reported. Here the letters A,B,C,D and E are replaced by the index 1 to 5. The high values on the diagonal show that the recognition process is very effective in recognizing all the letters (successful recognition 89%). Moreover, in the Figure 5(b), we report the false negatives, false positives, true positives and true negatives rates for each letter. Also in this case, we can observe rare false positives and false negatives, the good performance of the classifier in this case study shows that the system is very effective on intra user variability with a very small training set.



(a) Confusion matrix.

Letter	*FN-rate*	*FP-rate*	*TP-rate*	*TN-rate*
1*'A'*	0.0123	0.0000	1.0000	0.9877
2*'B'*	0.0247	0.0526	0.9474	0.9753
3*'C'*	0.0256	0.1818	0.8182	0.9744
4*'D'*	0.0370	0.1053	0.8947	0.9630
5*'E'*	0.0380	0.1905	0.8095	0.9620

(b) Classification results.

Fig. 5. Results for letters recognition case study.

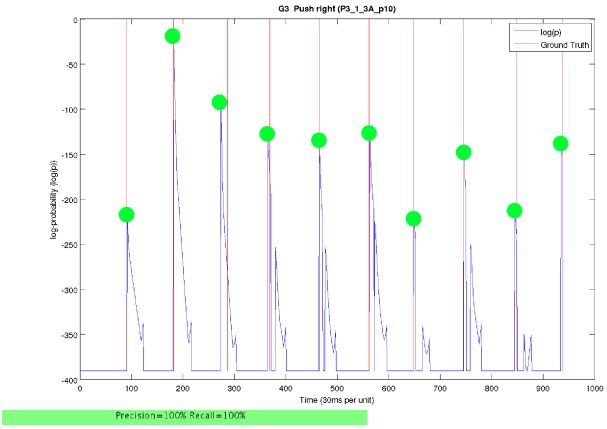
MSRC-12 case study results: As for the second case study, in Figures 6(a) and 6(b) we report the output log-probability for two gestures sequences, the *G3 - Push right* and *G4 - Googles*, respectively. The vertical (red) lines are the ground truth (e.g., when the gesture is considered performed by the user). The (blue) curve plot is the output log-probability for the given model (e.g., the confidence of a gesture to be recognized). When the probability reaches a peak and goes above a given threshold, the gesture is considered as recognized (see the green circles on top of the peaks). Here, we can observe that the peek is very close to the ground truth, therefore the gesture can be considered as successfully recognized for each gesture instance.

The Table I reports the results for the MSRC-12 case study in terms of accuracy, precision, and recall.

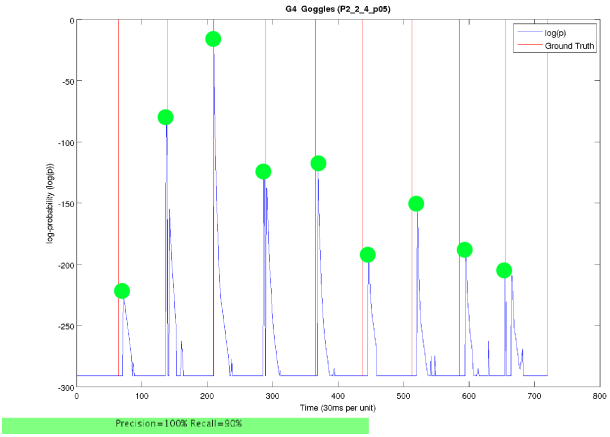
The precision is obtained as the ratio between TP classification (true positives, i.e. correct results) and the sum of TP and FP (false positives, i.e. unexpected results). On the other way recall is obtained measuring the ratio between TP and the sum of TP and FN (false negatives, i.e. missing results) classification. Accuracy is then computed as the ratio of the sum of TP and TN over the sum of TP, TN (true negatives, i.e. correct absence of results) FP and FN.

Also in this case, the results seem to confirm a good performance of the proposed method in this case study. This result is comparable to other results obtained with standard techniques in literature [6].

Human-Robot interaction case study results: Finally, we tested the system at work in a human-robot interaction context. Our experimental trials consist of a robotic manipulator cooperating with a human operator in simple tasks. In our setting, the robotic manipulator is close to a small table,



(a) Output log-probability for a *G3 Push right*.



(b) Output log-probability for a *G4 Goggles*.

Fig. 6. Log-probability for the MSRC-12 case study. On the horizontal axis is reported the time, while on the vertical axis is reported the log-probability of the recognition process. The vertical red lines represent the ground truth (when the gesture is performed by the user) while the blue line shows the log-probability of the recognition at a given time interval. Each green circle represents the local maxima of the log-probability. When the green circle is high it means that the confidence of the gesture recognition is high.

on which three differently coloured cups are placed; the red cup represents the target. The test we proposed is a game in which the participant has to drive the robotic arm towards the target object (a red cup) as many times as possible in a predefined amount of time (2 minutes). The selected testers have been explained which gestures they could use to interact with the robotic arm. 10 subjects participated in this experiment: 5 students and 5 PhD students, 6 males and 4 females. We evaluated the system considering both quantitative and qualitative performance. The quantitative measures are related to effectiveness and efficiency of the interactive system.

As far as qualitative performance are concerned, our aim was to evaluate the naturalness of the interaction from the operator's point of view. For this purpose, we defined a questionnaire to be filled by the tester after the overall session of tests. The questionnaire is inspired by the HRI questionnaire adopted in [3]. Its aim is to gain information

TABLE I

RESULTS FOR THE MSRC-12 CASE STUDY

Gesture	*Accuracy*	*Precision*	*Recall*
G1 lift outst. arms	0.7518	0.9285	0.7506
G2 Duck	0.7800	0.9545	0.7767
G3 Push right	0.8672	0.9759	0.8664
G4 Goggles	0.8015	0.9653	0.7993
G5 Wind it up	0.8534	0.9693	0.8656
G6 Shoot	0.7582	0.9591	0.7627
G7 Bow	0.8250	0.9675	0.8332
G8 Throw	0.8739	0.9705	0.8804
G9 Had enough	0.7937	0.9491	0.7824
G10 Change weapon	0.8273	0.9831	0.8174
G11 Beat both	0.6781	0.9280	0.6398
G12 Kick	0.7893	0.9642	0.8064
Average	0.8000	0.9596	0.7984

	Mean	STD	Min	Max
Failures	1.60	1.20	0	4
Successes	8.11	0.99	7	10

(a) Quantitative analysis.

	Mean	STD	Min	Max
Q1. (Competence)	1.80	0.87	1	4
Q2. (Ease of use)	3.40	0.66	2	4
Q3. (Naturalness)	3.50	0.92	2	5
Q4. (Satisfaction)	2.90	0.94	1	4
Q5. (Learning)	2.60	0.66	2	4

(b) Qualitative analysis.

Fig. 7. Experimental results for the HRI case study.

about subjects perception when interacting with the robotic arm. All questions presented may be answered with a grade from 1 to 5.

In particular, we consider two main sections for the users qualitative evaluation 1) a *Specific Information* section, where questions concern respectively *a*) user competences; and the feeling of easy of use; *b*) a *General Feelings* section, asking for naturalness, satisfaction and easy of learning feelings of the interaction [7]. In Figure 7, we report both quantitative and qualitative results. As for quantitative results, we measured the number of successes/failures collected by each tester reporting the average values, standard deviation, min, and max. Although the majority of the subjects was not used to interact with robotic systems, they could accomplish the proposed task after few attempts (Figure 7(a)). This is also confirmed by the qualitative results (Figure 7(b)-Q1), indeed the users considered both the naturalness (Figure 7(b)-Q3) and the ease of use to be satisfactory (Figure 7(b)-Q4). Learning how to use the system was also reported to be easy for the subjects (Figure 7(b)-Q5). Regarding the ease of use of the system the subjects reported a satisfaction level above the average (Figure 7(b)-Q4). This result is consistent with the difference the mean number of successes and the mean number of failures (Figure 7(a)). The realized system has been globally judged to be intuitive and satisfactory from the users' point of view.

IV. CONCLUSIONS

In this work, we presented a novel method for continuous gesture recognition that should support a natural and flexible human-robot social interaction. The proposed approach presents several advantages both during the training and the recognition phases. During the first phase, the available training set can be very small, hence the user can perform a very limited number of gesture samples to introduce new gestures. As for the second phase, a continuous recognition process enables the system to keep multiple hypothesis about the gesture switching from one interpretation to another depending on the context. This flexible and light process is possible because the bayesian classifier used in this work requires minimum computational resources and thus multiple gestures can be tracked in real-time. This continuous gesture recognition process allows us to face and resolve the ambiguities according to the interactive context in that enhancing the overall recognition system robustness and naturalness. The effectiveness of the recognizer has been tested in two standard case studies, while the flexibility and naturalness of the interaction has been discussed considering a simple HRI task. As a future work, we plan to investigate the system performance in a more sophisticated social interaction scenario considering full body gesture recognition.

REFERENCES

[1] A. F. Bobick and A. D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(12):1235–1337, 1997.

[2] J. Davis and M. Shah. Visual gesture recognition. *Vis., Image Signal Process.*, 141:101–106, 1994.

[3] M. Duguleana, F. G. Barbuceanu, and G. Mogan. Evaluating human-robot interaction during a manipulation experiment conducted in immersive virtual reality. In *Proc. of International Conference on Virtual and Mixed Reality: new trends - Part I*, pages 164–173. Springer-Verlag, 2011.

[4] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin. Instructing people for training gestural interactive systems. *ACM Conference on Human Factors in Computing Systems*, 2012.

[5] P. Hong, M. Turk, and T. S. Huang. Gesture modeling and recognition using finite state machines. pages 410–415, 2000.

[6] M. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. *IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2466–2472, 2013.

[7] S. Iengo, A. Origlia, M. Staffa, and A. Finzi. Attentional and emotional regulation in human-robot interaction. *In proceedings of the 21st IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN2012*, 2012.

[8] F. Nicolas and E. Rivals. Hardness results for the center and median string problems under the weighted and unweighted edit distances. *Discrete Algorithms*, 3(24):390 – 415, 2005.

[9] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–285, 1989.

[10] F. Samaria and S. Young. Hmm-based architecture for face identification. *Image Vis. Comput.*, 12:537–543, 1994.

[11] Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. *Rep. TR-375*, 1995.

[12] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hmm. *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, pages 379–385, 1992.

[13] M. S. Yang and N. Ahuja. Recognizing hand gesture using motion trajectories. *Proc. IEEE CS Conf. Comput. Vis. Pattern Recogn.*, 1:466–472, 1998.

[14] T. Yang and Y. Xu. Hidden markov model for gesture recognition. *CMU-RI-TR-94*, 10.