

# Fast Gesture Recognition with Multiple Stream Discrete HMMs on 3D Skeletons

Guido Borghi, Roberto Vezzani and Rita Cucchiara  
DIEF - University of Modena and Reggio Emilia  
Via P. Vivarelli 10, 41125 Modena, Italy  
Email: {name.surname}@unimore.it

**Abstract**—HMMs are widely used in action and gesture recognition due to their implementation simplicity, low computational requirement, scalability and high parallelism. They have worth performance even with a limited training set. All these characteristics are hard to find together in other even more accurate methods. In this paper, we propose a novel double-stage classification approach, based on Multiple Stream Discrete Hidden Markov Models (MSD-HMM) and 3D skeleton joint data, able to reach high performances maintaining all advantages listed above. The approach allows both to quickly classify pre-segmented gestures (offline classification), and to perform temporal segmentation on streams of gestures (online classification) faster than real time. We test our system on three public datasets, *MSRAction3D*, *UTKinect-Action* and *MSRDailyAction*, and on a new dataset, *Kinteract Dataset*, explicitly created for Human Computer Interaction (HCI). We obtain state of the art performances on all of them.

## I. INTRODUCTION

Capturing and understanding people behavior through vision and RGB-D data is a popular and challenging research topic. Since the area is very broad, the proposed systems are often specialized, according with the context, the specific application goals and the type of observed human activity. Depending on the task, the temporal granularity and the semantic abstraction level, different terms have been adopted to describe atomic body movements such as posture, gesture, action, interaction, behavior, and so on. In this paper, we focus on the recognition of *dynamic body gestures for explicit Human-Computer Interaction* (HCI), that can be defined as follows:

- **Dynamic**: the target gesture requires a movement; thus we neglect static postures (e.g., *sitting*, *reading a book*);
- **Body**: the target gesture is potentially performed using the whole body; thus we discard too local gestures such as finger movements or facial expressions;
- **Gestures**: a gesture is a well-defined and time-limited body movement; continuous actions such as *running*, *walking* are not considered;
- **Explicit HCI**: we focus on gestures provided by a user which has spontaneously decided to interact with the system; thus, the gesture recognition subsumes a corresponding reaction or feedback at the end of each gesture.

Different methods have been proposed in the past to address this problem; currently, the most common solutions adopted in real-time applications include a 3D sensor and a classifier based on Dynamic Time Warping (DTW) or a Hidden Markov

Model (HMM) working on body joint positions. Even if more complex solutions have been proposed, DTW and HMM based systems have a wider diffusion thanks to their simplicity, low computational requirements and scalability; moreover, even a limited training set allows to reach worth performance. Although the recent release of cheap 3D sensors, like *Microsoft Kinect*, bring up new opportunities for detecting stable body points, developing a gesture recognition solution with characteristics of efficiency, efficacy and simplicity together is still far to be accomplished. Here we propose a new framework, which performs an on-line *double-stage Multiple Stream Discrete Hidden Markov Model* (double-stage MSD-HMM) to classify gestures. In the HCI context we address, users are in front of the acquisition and have the (intentional) need to exchange information with the system through their natural body language. Differently from other works that use standard HMMs and are focused on defining and detecting *ad hoc* feature sets to improve performances, we focus our attention on the whole pattern recognition flow, from the new HMM architectural setting to its implementation details, in order to obtain a real time framework with good performances on speed and classification.

## II. RELATED WORKS

Thanks to the spreading of low cost 3D sensors, the research on gesture recognition from RGB-D data has grown interest. In particular, the availability of almost accurate streams of body joint 3D positions [1] allows the body posture or gesture recognition directly from skeleton data, without the need of working on the source depth map. 3D joint position could be affected by noise and estimation errors in presence of occlusions and depth maps intrinsically include richer information. Neural networks and deep architectures are able to extract good features and to correctly classify actions and gestures from the depth maps with impressive performances, when large annotated datasets are available [2]. Hybrid methods merge skeleton and depth features, in order to get both advantages and increase system performances. To this aim, solutions based on Random Forest [3] or multi-kernel learning in [4], [5] have been proposed. However, although methods based on skeleton only may be affected by errors and noise, their computational requirements are usually less demanding and more suitable for real-time systems. Thus, in this work we focus on a solution based on 3D joint positions only. Following

the same assumption, Evangedilis *et al.* in [6] proposes a local, compact and view-invariant skeletal feature to encode *skeletal quad*. Chaudhry *et al.* in [7] proposes a bio-inspired method to create 3D discriminative skeletal feature. In [8] action recognition is performed through skeletal pose-based features, built on location, velocity and correlation joint data. Xia *et al.* in [9] uses histograms of 3D joints locations to perform classification through discrete HMMs. HMM models has been widely used due to their low computational load and parallelization capabilities. Also [10] suggests an approach based on HMMs by modelling the state emission probability with neural networks, which somehow nullify the advantages of HMM models. Finally, very few works [11], [12] have been proposed in the past with discrete weighted multiple stream HMMs, but only in facial expression and handwriting recognition tasks respectively (no gesture recognition task).

### III. OFFLINE GESTURE CLASSIFICATION

The core part of the proposal is a probabilistic solution for the classification problem of a pre-segmented clip, containing a single gesture (offline classification). Given a set of  $C$  gesture classes  $\Lambda = \lambda^1 \dots \lambda^C$ , we aim at finding the class  $\lambda^*$  which maximizes the probability  $P(\lambda|O)$ , where  $O = \{o_1 \dots o_T\}$  is the entire sequence of frame-wise observations (i.e. the features). A set of HMMs, each trained on a specific gesture, is the typical solution adopted for this kind of classification problems [13]. The classification of an observation sequence  $O$  is carried out selecting the model  $\lambda^*$  whose likelihood is highest. If the classes are a-priori equally likely, this solution is optimal also in a Bayesian sense.

$$\lambda^* = \arg \max_{1 \leq c \leq C} [P(O|\lambda^c)] \quad (1)$$

If the decoding of the internal state sequence is not required, the standard recursive forward algorithm for HMMs with the three well known initialization, induction and termination equations can be applied:

$$\begin{aligned} \alpha_1(j) &= \pi_j b_j(o_1), 1 \leq j \leq N \\ \alpha_{t+1}(j) &= \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1}) \\ P(O|\lambda) &= \sum_{j=1}^N \alpha_T(j) \end{aligned} \quad (2)$$

where  $\pi = p(q_i = s_i)$  is the initial state distribution,  $a \in A$ , matrix  $A$  describes the transition  $p(q_t = s_i | q_{t-1} = s_i)$  for hidden states  $S$  ( $q_t$  is the current state),  $b_j(o)$  depends on the type of the observation and defines the emission state probabilities.

A common solution is based on Gaussian Mixture Models (GMMs) learned on a feature set composed of a set of continuous values [14]. The term  $b_j(o_t)$  of Eq. 2 would be approximated as:

$$b_j(o_t) = \sum_{l=1}^M c_{jl} \mathcal{N}(o_t | \mu_{jl}, \Sigma_{jl}) \quad (3)$$

where  $M$  is the number of Gaussian components per state,  $\mu_{jl}$  and  $\Sigma_{jl}$  are the Gaussian parameters, and  $c_{jl}$  are the mixture weights.

### A. Feature set

We assume to already have the body joint 3D coordinates as input [1]. In this work, we exploited a simple feature set directly derived from the joint stream, discriminative enough to obtain reasonable classification rates. Additional features may be included without changing the classification schema, but this may go to the detriment of the computational complexity and to the overall efficacy due to the curse of dimensionality. Thus, only nine features are extracted for each selected body joint  $K_i$ . Given the sequence of 3D positions of the joint  $K_i^t = (x_i^t, y_i^t, z_i^t)$ , we define  $o_t(i) = \{o_t^1(i) \dots o_t^9(i)\}$  as:

$$\begin{aligned} o_t^1(i), o_t^2(i), o_t^3(i) &= x_i^t - \hat{x}_i^t, y_i^t - \hat{y}_i^t, z_i^t - \hat{z}_i^t \\ o_t^4(i), o_t^5(i), o_t^6(i) &= x_i^t - x_i^{(t-1)}, y_i^t - y_i^{(t-1)}, z_i^t - z_i^{(t-1)} \\ o_t^7(i), o_t^8(i), o_t^9(i) &= x_i^t - 2x_i^{(t-1)} + x_i^{(t-2)}, \\ & y_i^t - 2y_i^{(t-1)} + y_i^{(t-2)}, z_i^t - 2z_i^{(t-1)} + z_i^{(t-2)} \end{aligned} \quad (4)$$

where  $\{o^1 \dots o^3\}$  are the joint positions with respect to a reference joint  $k$  selected to make the feature set position invariant;  $\{o^4 \dots o^6\}$  and  $\{o^7 \dots o^9\}$  are the speed and acceleration components of the joint respectively. This approach is inspired by [8], [3]. A linear normalization is applied to normalize the feature values to the range of  $[-1, 1]$ . Selecting a subset (or the whole set) of  $G$  body joints, the complete feature set  $o_t$  for the  $t$ -th frame is obtained as a concatenation of  $D = 9 \cdot G$  features. Thanks to the limited dependencies among the features, fast and parallel computation is allowed.

### B. Multiple Stream Discrete HMM

The computation of exponential terms included in Eq. 3 is time-consuming. Moreover, the Gaussian parameters  $\mu_{jl}, \Sigma_{jl}$  may lead to degenerate cases when learned from few examples. In particular, in the case of constant features the corresponding  $\Sigma_{jl}$  values becomes zero. Even if some practical tricks have been proposed in these cases [13], overall performances generally decrease when few input examples are provided in the learning stage. For these reasons, we propose to adopt discrete HMM. All continuous observations  $O$  are linearly quantized to  $L$  discrete values  $l \in [1 \dots L]$ . The adoption of discrete distributions to model the output probability solves the above mentioned numerical issues. In addition, to improve the generalization capabilities of HMM with limited samples for each state, we adopted a set of  $D$  independent distributions –one for each feature item– called *streams*; the term  $b_j(o_t)$  of Eq. 3 is replaced with the following one:

$$b_j(o_t) = \prod_{d=1}^D (h_j^d(o_t^d))^{\alpha_d} \quad (5)$$

where  $(h_j^d(o_t^d))$  is the emission probability of the discrete value  $o_t^d$  in the  $j$ -th state and  $\alpha_d$  are weighting terms. The observation of each stream is thus *statistically independent* from the others given the current state  $q_t$ .

The weight coefficients  $\alpha_d$  may be used to take into account the different classification capability of each joint and each feature component (see Eq. 4).

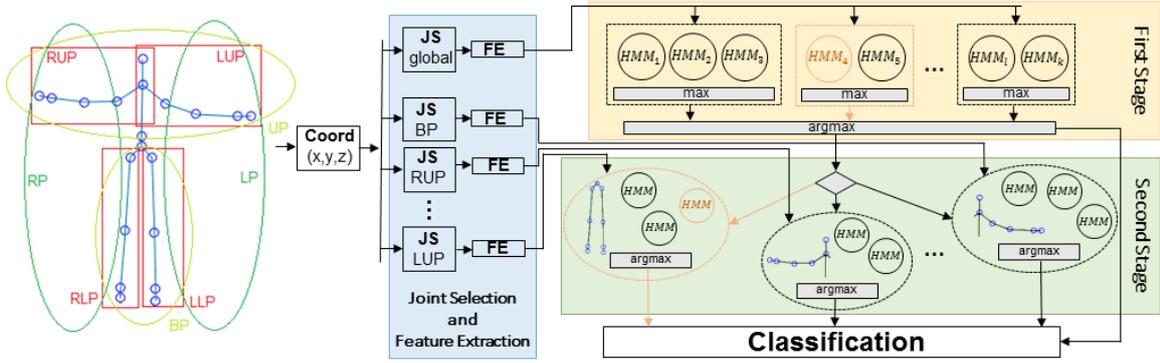


Fig. 1. Block-diagram overview of the proposed method.

#### IV. DOUBLE-STAGE CLASSIFICATION

The previous described approach does not discriminate if there are some parts of the body that are more significant for a given gesture, since the same feature set is provided as input to all the HMMs. For example, the gesture “hello” can be better recognized if the hand’s joints are analyzed alone without other distracting joints. However, the responses HMM with different feature sets becomes not comparable. Following this observation, a *double-stage* classification system based on MSD-HMM is proposed as outlined in Fig. 1. Gestures are grouped into sets depending on the sub set of the interested body joints. The first classification stage recognizes the gesture group (i.e., discovers which part of human body is most involved in the gesture), while the second stage provides the final classification among the gestures in the selected group. The MSD-HMMs of the first stage work on a global subset of joints extracted from the whole body, while the MSD-HMMs of the second stage are more specific on the body part involved. In particular, the global subset of joints (used to compute the features of Eq. 4) contains the left and right foot, the left and right hand and the head. Other joints, like shoulders or knees, are instead used in the second stage. Gesture groups of the second-stage correspond with the partition reported on the left part of Fig.1. Four local-areas (i.e., Right Upper Part (RUP), Left Upper Part (LUP), Right Lower Part (RLP), Left Lower Part (LLP)) are firstly defined. Then, four additional macro-areas are created as combinations of local-areas (i.e., Upper Part (UP), Bottom Part (BP), Right Part (RP) and Left Part (LP)). Eight gesture clusters are correspondingly defined based on the main body part involved. A different set of stream weights  $\alpha_d$  (see Eq. 5) is computed for each HMM, using the average motion of each joint: the high the motion of a body part in a gesture is, the high is the corresponding stream weights.

#### V. ONLINE TEMPORAL SEGMENTATION AND CLASSIFICATION OF GESTURES

Differently from offline gesture classification, online recognition requires to estimate the most likely gesture currently performed by the monitored subject, given only the observations until now. The observed sequence may contain more

instances and the current gesture may be in progress. The original proposed system is able to detect gestures on-line, overtaking classical static approaches that are characterized by strong prior hypothesis. For example, common *sliding window* methods implicitly apply a strong constraint on the average and maximum duration of each gesture. Since HCI is often characterized by a high variability on the gesture duration and noise, the proposed solution is not based on fixed-length rigid windows. Using HMM, the temporal evolution of a gesture is correlated with the hidden state probability. In particular, exploiting the *left-right* transition model [13] the first and last state of the chain can be exploited to detect the temporal boundaries of each gesture. The following rules are applied to the first-stage classification layer of Fig. 1:

- **Beginning detection:** the beginning of a gesture candidate is detected by analyzing the first hidden state of each HMM. The adoption of a left-right transition model is mandatory. The most likely state of HMM will be the first one during non-gesture times, while the following states will be activated once the gesture starts. A voting mechanism is exploited, counting how many HMMs satisfy the following condition:

$$\phi(HMM_k) = \begin{cases} 0 & \alpha_t(1) = \left[ \sum_{i=1}^N \alpha_t(i) a_{i1} \right] b_1(o_t) \geq th \\ 1 & \alpha_t(1) = \left[ \sum_{i=1}^N \alpha_t(i) a_{i1} \right] b_1(o_t) < th \end{cases} \quad (6)$$

where  $N$  is the total number of hidden states in HMMs and  $b$  is defined in Eq. 5. If a large number of models satisfies Eq. 6, a gesture is performing. The threshold  $th$  on  $\alpha_t(1)$  checks the probability to be still at the instant  $t$  in the first hidden state.

- **End detection:** if a gesture is currently performed (as detected by the previous rule) the most likely gesture is computed at each frame as in Eq. 1. A probability distribution analysis of the last state of the corresponding HMM is performed in order to detect the end of the gesture:

$$\alpha_t(N) = \left[ \sum_{i=1}^N \alpha_t(i) a_{iN} \right] b_j(o_t) \geq th \quad (7)$$

where  $N$  is the total number of hidden states and  $th$  is a threshold on  $\alpha_t(N)$ , the probability to be at the instant  $t$  in the last ( $N$ -th) state after observations  $O(t_s \dots t_e)$ .

- **Reliability Check:** this rule is applied to filter out false candidates (non valid or incomplete gestures, for example). The sequence of hidden state probabilities obtained with the selected HMM model on the observation window is analyzed. Starting from the set  $S$  of hidden states, the subset  $\hat{S} \subset S$  is defined as follows:

$$\hat{S} = \{s_j | \forall t \in [t_s, t_e], \alpha_t(j) \geq th\} \quad (8)$$

The following checks are performed to validate the candidate gesture:

$$\begin{aligned} \#\hat{S} &\geq \frac{2}{3}N \\ S_{N-1} &\in \hat{S} \end{aligned} \quad (9)$$

where  $\#\hat{S}$  is the cardinality of the set  $\hat{S}$ . The previous checks guarantee that at least  $\frac{2}{3}$  of hidden states (included the second to last) are visited. A state is defined as visited if there is an instant  $t$  characterized by a high probability ( $th > 0.90$ ) to be in that state.

Once the first stage classification layer provides a valid temporal segmentation and the corresponding simultaneous gesture classification, the second-stage layer is exploited to refine the classification.

## VI. EXPERIMENTAL RESULTS

We test our system on three public and famous datasets, *MSRAction3D*, *MSRDailyActivity3D* and *UTKinect-Action*, as well as on a new custom dataset (*Kinteract Dataset*) we developed for HCI.

### A. MSRAction3D Dataset

MSR Action3D dataset [15] contains 20 action classes performed by 3 subjects: *high arm wave*, *horizontal arm wave*, *hammer*, *hand catch*, *forward punch*, *high throw*, *draw x*, *draw tick*, *draw circle*, *hand clap*, *two hand wave*, *side-boxing*, *bend*, *forward kick*, *side kick*, *jogging*, *tennis swing*, *tennis serve*, *golf swing and pickup & throw*. In total there are 567 action instances. This dataset fits our method due to the absence of human-object interaction and it contains only 3D joint position. This is one of the most used dataset for human action recognition task, but in [16] are reported some issues about validation methods: the total number of samples used for training and testing is not clear, because some 3D joint coordinates are null or with highly noise. To avoid ambiguity, the complete list of valid videos we used is publicly available<sup>1</sup>. The validation phase has been carried out following the original proposal by Li *et al.* [15]. Three tests are performed: in the first two tests, 1/3 and 2/3 of the samples are used for training and the rest for the testing phase; in the third test half of the subjects are used for training (subjects number 1, 3, 5, 7, 9) and the remainder for testing (2, 4, 6, 8, 10).

<sup>1</sup><http://imagelab.ing.unimore.it/hci>

### B. UTKinect-Action Dataset

UTKinect-Dataset [9] contains 10 action classes performed by 10 subjects: *walk*, *sit down*, *stand up*, *pick up*, *carry*, *throw*, *push*, *pull*, *wave hands*, *clap hands*; each subject performed every action twice. There are 200 action sequences. This dataset is challenging due to high intra-class variations and the variations in the view point. Moreover, because of the low frame rate and the duration of some actions, some sequences are very short. The validation method proposed in [9] has been adopted, using a leave one sequence out cross validation.

### C. MSRDailyActivity3D Dataset

DailyActivity3D dataset [5] contains daily actions captured by a Kinect device. There are 16 activity classes performed twice by 20 subjects: *drink*, *eat*, *read book*, *call cellphone*, *write on a paper*, *use laptop*, *use vacuum cleaner*, *cheer up*, *sit still*, *toss paper*, *play game*, *lay down on sofa*, *walk*, *play guitar*, *stand up*, *sit down*. Generally, each subject performs the same activity standing and sitting on the sofa. There is a total of 320 long activity sequences. Since our proposed method is specifically conceived for dynamic gestures (see Sec. I), still actions (e.g., read book, sit still, play game and lay down on the sofa) have been removed during the evaluation. We follow the cross-subject test proposed in [5], [17].

### D. Kinteract Dataset

In addition, we collect a new dataset publicly available<sup>1</sup>, which has been explicitly designed and created for HCI. Ten common types of HCI gestures have been defined: *zoom in*, *zoom out*, *scroll up*, *scroll down*, *slide left*, *slide right*, *rotate*, *back*, *ok*, *exit*. For example, these gestures can be used to control an application using the *Kinect* sensor as a natural interface. Gestures are performed by 10 subjects for a total of 168 instances and are acquired by standing in front of a stationary *Kinect* device. Only the upper part of the body (from shoulders to hands) is involved. Each gesture is performed with the same arm by all the subjects (despite they are left or right handed). This dataset allows to highlight the advantages of our solution in a real world context.

### E. Experiments

The output of the first-stage HMM can be directly used for classification (using *argmax* instead *max* in Fig. 1). We report the corresponding performance as reference for the final double-stage one. The number of HMM hidden states is set to 8. The number  $L$  of quantization levels (see Sec. III-A) has been empirically selected on the *MSRAction3D* dataset. As reported in Fig. 3, the best value is  $L = 10$ . Finally, stream weights are equally initialized to 1 by default.

Table I contains an internal comparison of the system on the *MSRAction3D* dataset, based on cross subjects evaluation method. The complete system is compared with baseline solutions where the feature normalization (FN) and the stream weight (WMS) steps are not performed.

Table II reports the performance of state of the art methods on the same dataset. Results show that our method is in line

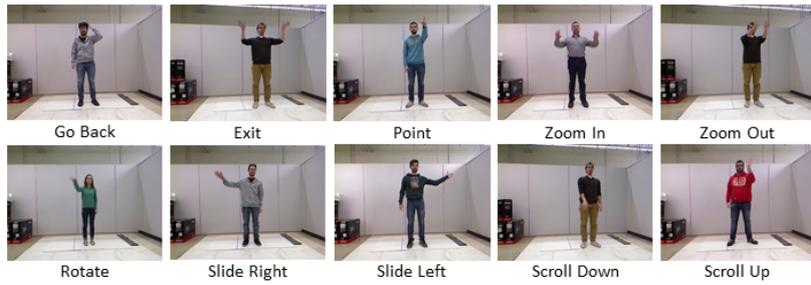


Fig. 2. Action classes from our *Kinteract Dataset*. Classes are explicitly created for Human Computer Interaction.

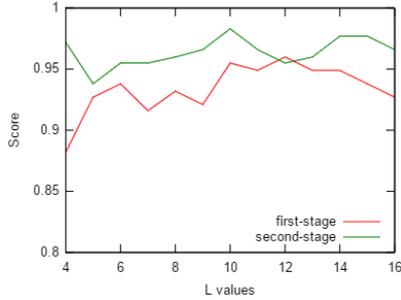


Fig. 3. System performance w.r.t. quantization levels

TABLE I  
SYSTEM PERFORMANCE

MSRAction3D		
System Part	First Stage	Second Stage
Base Method	0.739	0.802
Base + FN	0.845	0.873
Base + FN + WMS	0.861	<b>0.905</b>

FN: Feature Normalization; WMS: Weighted Multiple Streams

with the state of the art, where methods based on skeleton only are listed. The approach [8] provides better results but is really much more expensive in terms of feature extraction and classification time, as described below.

Results on *UTKinect-Action* dataset are reported in Table III: we run the experiment 20 times and we get the mean performance; however, the best accuracy obtained in a run

TABLE II  
RESULTS ON THE MSRAction3D DATASET

Methods	Accuracy		
	1/3	2/3	cross
HOJ3D [9]	0.962	0.971	0.789
HMM + DBM [10]	-	-	0.820
EigenJoints [18]	0.958	0.977	0.823
HMM + GMM (our implementation)	0.861	0.929	0.825
Actionlet Ensemble (skeleton data) [5]	-	-	0.882
Skeletal Quads [6]	-	-	0.898
LDS [7]	-	-	0.900
Cov3DJ [19]	-	-	0.905
Our	0.943	0.983	0.905
FSFJ3D [3]	-	-	0.909
KPLS [8]	-	-	0.923

TABLE III  
RESULTS ON THE UTKINECT-ACTION DATASET

Methods	Accuracy
DSTIP+DCSF[3]	0.858
FSFJ3D (skeleton data)[3]	0.879
SNV [20]	0.889
Our	0.895
HOJ3D [9]	0.909

is 0.965. Table IV reports results for the *MSRDailyAction* dataset. Even if the comparison is not completely fair since we considered a subset of gestures and we only use skeleton data, the reported performance confirms the generalization capability of our method as well as its efficacy on long and generic actions. The proposed system has an overall accuracy of 0.974 on the *Kinteract* dataset performing a cross subject test. Finally, we test the online temporal segmentation. Instances belonging to *MSRAction3D* and *Kinteract* dataset are merged in a continuous stream. Corresponding results are reported in Table V. The threshold for state analysis probability distribution is  $th = 0.9$  (see eq. 6). A temporal segment is valid only if has a Intersection over Union (IoU) with ground truth instance larger than overlap threshold ( $\sigma = 0.5$ ); the ground truth instance with greater IoU defines the current class action that has to be classified. Results show that our original temporal segmentation method can be used in a real world framework. We implemented the proposed system in C++ and tested on a computer with Intel i7-4790 (3.60GHz) and 16 GB of RAM. The framework is able to extract and calculate features, perform feature quantization, evaluate stream weights and classify a single action with an average time of  $4.4 \cdot 10^{-2}s$  (tested on the *MSRAction3D* dataset with 20 HMM for each stage). Performance results with respect to the number of HMMs (for each stage) are reported in Fig. 4. The total number of HMM of each stage corresponds to the potential number of recognizable gesture classes. Results are normalized by the mean gesture length (41 frames). The complete online system runs at about 80.35 frames per second when trained on the *MSRAction3D* dataset. Its exploitation on real time systems is then guarantee, differently to other recent state of the art solutions [21], [22]; in particular, the feature extraction and classification time for a single action in [8] is about 300 ms.

TABLE IV  
RESULTS ON THE MSR DAILY ACTION DATASET

Methods	Accuracy
DTW [23]	0.540
Moving Pose [17]	0.738
HON4D [24]	0.800
Our	0.833
Actionlet Ensemble [5]	0.845

TABLE V  
RESULTS FOR ONLINE TEMPORAL SEGMENTATION

Dataset	Detection Rate	Recognition Rate
MSRAction3D	0.782	0.871
Kinteract	0.892	0.943

## VII. CONCLUSIONS

In this paper, we investigate and improve the use of on-line double-stage Multiple Stream Discrete HMM (MSD-HMM), that are widely used due to their implementation simplicity, low computational requirements, scalability and high parallelism, in action and gesture recognition fields. We also demonstrate that HMMs can be successfully used for gesture classification tasks with worth performances even with a limited training set. Thanks to a double-stage classification based on MSD-HMM, our system allows both to quickly classify and perform online temporal segmentation, with a great generalization capability. Results are in line with the state of the art on three public and challenging datasets. Before our method HMM were not able to compete in gesture classification task with other state-of-the-art methods. Moreover, a new gesture dataset is introduced, namely *Kinteract dataset*, which is explicitly designed and created for HCI.

## REFERENCES

- [1] 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 single depth images," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition 2011 (CVPR)*, June 2011, pp. 1297–1304.
- [2] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. Ogunbona, "Deep convolutional neural networks for action recognition using depth map sequences," *CoRR*, vol. abs/1501.04686, 2015.
- [3] Y. Zhu, W. Chen, and G. Guo, "Fusing spatiotemporal features and joints for 3d action recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2013.
- [4] S. Althloothi, M. H. Mahoor, X. Zhang, and R. M. Voyles, "Human activity recognition using multi-features and multiple kernel learning," *Pattern Recogn.*, vol. 47, no. 5, pp. 1800–1812, May 2014.
- [5] Y. Wu, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR '12, 2012, pp. 1290–1297.
- [6] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: human action recognition using joint quadruples," in *22nd International Conference on Pattern Recognition (ICPR)*, 2014.
- [7] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3d discriminative skeletal features for human action recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2013.
- [8] A. Eweiri, C. Cheema, Muhammed S. and Bauchhage, and J. Gall, "Efficient pose-based action recognition," in *12th Asian Conference on Computer Vision - Part V*, 2015, pp. 428–443.

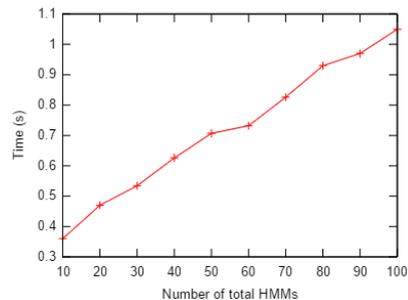


Fig. 4. Classification time w.r.t. the number of classes

- [9] L. Xia, C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2012, pp. 20–27.
- [10] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *Proc. of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Washington, DC, USA: IEEE Computer Society, 2014.
- [11] P. S. Aleksic and A. K. Katsaggelos, "Automatic facial expression recognition using facial animation parameters and multistream hmms," *IEEE Trans. Information Forensics and Security*, vol. 1, pp. 3–11, 2006.
- [12] J. Schenk, S. Schwrzler, and G. Rigoll, "Discrete single vs. multiple stream hmms: A comparative evaluation of their use in on-line handwriting recognition of whiteboard notes," in *Proc. 11th Intern. Conf. on Frontiers in Handwriting Recognition ICFHR 2000*. CENPARMI Concordia University Montreal, Canada, 2008, pp. 550–555.
- [13] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proc. of the IEEE*, vol. 77, 1989, pp. 257–286.
- [14] R. Vezzani, M. Piccardi, and R. Cucchiara, "An efficient bayesian framework for on-line action recognition," in *16th IEEE International Conference on Image Processing (ICIP)*, Nov 2009, pp. 3553–3556.
- [15] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2010*, 2010, pp. 9–14.
- [16] J. R. Padilla-López, A. A. Chaaoui, and F. Flórez-Revuelta, "A discussion on the validation tests employed to compare human action recognition methods using the MSR action3d dataset," *CoRR*, vol. abs/1407.7390, 2014. [Online]. Available: <http://arxiv.org/abs/1407.7390>
- [17] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection," in *IEEE International Conference on Computer Vision ICCV 2013*, 2013, pp. 2752–2759.
- [18] X. Yang and Y. Tian, "Eigenjoints-based action recognition using nave-bayes-nearest-neighbor," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. IEEE, 2012, pp. 14–19.
- [19] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations," in *Proc. of the 23th Int. Joint Conference on Artificial Intelligence*, ser. IJCAI '13, 2013, pp. 2466–2472.
- [20] X. Yang and Y. Tian, "Super normal vector for activity recognition using depth sequences," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 804–811.
- [21] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. BMVC*, 2009, pp. 124.1–124.11, doi:10.5244/C.23.124.
- [22] S. R. Fanello, I. Gori, G. Metta, and F. Odone, "One-shot learning for real-time action recognition," in *Proceedings of 6th Iberian Conference on Pattern Recognition and Image Analysis - IbPRIA 2013*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 31–40.
- [23] M. Müller and T. Röder, "Motion templates for automatic classification and retrieval of motion capture data," in *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006, pp. 137–146.
- [24] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *In CVPR*, 2013.