COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2016; **27**:321–329
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cav.1715

SPECIAL ISSUE PAPER

Dance performance evaluation using hidden Markov models

Sohaib Laraba* and Joëlle Tilmanne
TCTS Lab, Numediart Institute, University of Mons, Mons, Belgium

ABSTRACT

We present in this paper a hidden Markov model-based system for real-time gesture recognition and performance evaluation. The system decodes performed gestures and outputs at the end of a recognized gesture, a likelihood value that is transformed into a score. This score is used to evaluate a performance comparing to a reference one. For the learning procedure, a set of relational features has been extracted from high-precision motion capture system and used to train hidden Markov models. At runtime, a low-cost sensor (Microsoft Kinect) is used to capture a learner's movements. An intermediate step of model adaptation was hence requested to allow recognizing gestures captured by this low-cost sensor. We present one application of this gesture evaluation system in the context of traditional dance basics learning. The estimation of the log-likelihood allows giving a feedback to the learner as a score related to his performance. Copyright © 2016 John Wiley & Sons, Ltd.

KEYWORDS
gesture recognition; hidden Markov Models; interactive systems; maximum likelihood linear regression; performance evaluation

***Correspondence**
Sohaib Laraba, TCTS Lab, Numediart Institute, University of Mons, Mons, Belgium.
E-mail: sohaib.laraba@umons.ac.be

1. INTRODUCTION

In the present day, having an efficient human-computer interaction system is taking a significant role. Gesture recognition is considered as one of the important steps in order to achieve this objective. Gesture recognition systems have been successfully developed based on different methods like hidden Markov model (HMM), neural networks, finite-state machine (FSM), and template matching [1]. In most cases, the data used for training and the data to be recognized come from the same capturing system. Ideally, the data are captured by high-precision motion capture systems. These systems provide very precise data at framerate that reaches 200 frames per second (fps). However, such systems are very expensive and hence not suited for at-home use. In order to build an interactive dance learning system using motion capture technologies, we need to design a dance performance evaluation module. We present here an approach that allows at the same time to train robust models using high precision and to recognize gestures recorded in real time from a low-cost system that is cheaper and easier to obtain.

We are particularly interested on computing the similarity between the performed gesture and the reference gestures stored in the database and used for the training phase. As this will be illustrated later, the similarity score is useful in order to provide a feedback to the user and allows him to improve his performance. Gestures are considered to be multidimensional temporal curves representing relational features extracted from geometric relationships between different joints of users skeleton. This is inspired by Müllers work for analysis of motion data [2].

Our system is based on HMMs with an additional step of model adaptation using maximum likelihood linear regression (MLLR) procedure as discussed in [3]. This latter step is essential in order to have a sensor-dependent gesture recognition system. For decoding, Viterbi algorithm is used and allows to have a log-likelihood value, which will be transformed into a percentage score to be presented to the user.

This paper is structured as follows: First, we present a summary of related works. We describe the algorithm used for recognition with numerical results using real data from the second version of the Microsoft Kinect and a high-precision motion capture system (Qualisys[†]). Then, we show how we transformed the log-likelihood into a percentage score related to the performance. Finally, we present a typical use case for learning traditional dance basics.

[†]Qualisys Motion Capture System: http://www.qualisys.com/.

2. RELATED WORKS

Full body gesture recognition has been performed with different approaches, and in each one, different motion features have been used. In a first stage, we cite recent feature extractors used to represent a motion sequence, then we briefly summarize gesture classification approaches. In the last part, we describe some techniques for gesture evaluation and scoring.

2.1. Feature Extraction

Different feature extraction methods have been proposed, based on skeletal information of the body. In some works, like in [4], 3D angular values, in addition to their first and second derivatives (velocity and acceleration), were taken into account in order to model stylistic gait sequences. In [5], a set of features is computed by calculating the Euclidean distance between every pair of 3D joints in the current frame and the distances between the joints in the current frames and the ones in the previous frame. To capture the overall dynamics of body movement, similar distances are computed between the current frame and a neutral pose. The neutral pose is computed by averaging the initial skeletons of all action sequences. Each individual feature value was clustered into five groups via k-means and replaced with a five-bit binary vector. Müller *et al.* [6] introduced geometric features that are a class of Boolean features expressing geometric relations between certain body points of a pose, for example, whether the right foot lies in front of or behind the plane spanned by the left foot, the left hip joint, and the center of the hip. Such geometric features are robust to spacial variations and allow the identification of logically corresponding events in similar motions. Other methods learn a dictionary of key poses and represent an action sequence in term of these key poses. Ofli *et al.* [7] used a histogram of motion words (HMW) where a set of 3D locations representing the most informative joints are clustered into K poses (or motion words) by using K-means. An action sequence is represented by counting the number of detected motion words.

2.2. Gesture Classification

To deal with the temporal warping that affects motion sequences and ensure equal lengths, dynamic time warping (DTW) has been used in many works like in [6], [8], and [9]. Classification is then performed by K-nearest neighbor. FSMs have been efficiently employed in modeling human gestures [10], and they were combined with support vector machines in [11]. A lot of efforts [4,12,13] have used HMMs for modeling body motion time series. Four real-time decoding algorithms based on HMMs have been presented in [4] for stylistic gait recognition and following. These methods are based on Viterbi algorithm for decoding but each one uses a different approach. The algorithms are evaluated on their ability to recover the progression over time in real time. Bevilacqua *et al.* [12] developed a learning strategy based on a single recorded example. Their system outputs the time progression index and the likelihood values that are used for decoding. A major limitation of this technique is the large number of states that can hinder the real-time computation. Our system is based on HMMs containing a fixed number of states and trained on higher number of samples in order to model the variability of the gestures. HMMs integrate both the time and the stylistic variability of the motion in their modeling thanks to their topology. Our approach is described in Section 3 and allows, with an adaptation technique, to have a sensor-dependent gesture recognition system, which means that the data used for decoding can be different than the one used for training (from different sensor for example). For decoding, a standard Viterbi algorithm is used.

2.3. Performance Evaluation

In order to learn and improve dance steps, having a score that evaluates the performance can be very helpful. For evaluation of a dancer's performance, Kitsikidis *et al.* [14] performed evaluation by computing distances between knees and ankles of the learner and a reference dancer and attributed a final score based on these distances. However, this method does not deal with temporal warping of the reference dancer gestures and the learner ones. In [6], Müller *et al.* dealt with this issue by using DTW and creating a binary template of the reference gesture, and then the similarity is measured by computing a distance between the template and the gesture to compare. Bloom *et al.* [15] sum occurrences of true positives, false negatives, and false positives along an event timeline to produce an F1 score. Chan *et al.* [16] proposed a dance training system based on motion capture and virtual reality technologies. The student's motions are captured by a high-precision motion capture system when he tries to imitate a teacher's movements. Their system computes the differences between the sequences of the student and the teacher using DTW and Euclidean distance and provides a score for each joint and an overall score for the whole performance. The features that are used to compute these distances are positions, velocities, or angles. However, the data used by their system are provided by a high-precision motion capture system, which is very expensive and needs special setup, which make it difficult to have this system for home applications. In our work, a gesture is evaluated using HMMs. When decoding a gesture using Viterbi algorithm, we output the log-likelihood, which is interpreted and translated to a percentage score. The approach is detailed in Section 3.5.

3. GESTURE MODELING

As described generally in machine learning techniques, a gesture recognition system includes two procedures, learning and decoding. As mentioned previously, our system is developed under the constraint that the data used for decod-

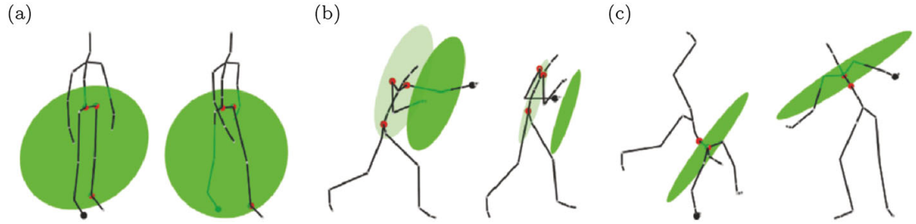


Figure 1. Relational features describing geometric relations between body points of a body pose that are indicated by red and black markers [2].

ing are different than the one used for learning. Hence, an additional procedure of adaptation is required.

The HMM model used for gesture recognition was trained using hidden Markov model toolkit (HTK [17]). The training dataset was recorded using a high-precision motion capture system (Qualisys). An expert of a traditional dance from the south of Belgium (Walloon Dance) was recorded performing different basic steps. These steps are as follows: Maclotte Base, Passepied Base, Passepied Fleuret, and Back step. The expert was recorded also using a low-cost device (Microsoft Kinect V2) in order to select the features that are not strongly affected by the change of motion capture system. The recorded sequences are defined as series of forward and backward steps and the dancer do not have to turn. This kind of gestures facilitated our study because the Kinect device requires a specific setup in order to work correctly. In fact, the user needs to be facing the Kinect permanently in order to be well tracked. The Qualisys records motion at a framerate of 177 fps. Each frame contains 3D position of 68 markers placed on the body = 204 values per frame. The motion data were filtered to 30 fps, and the 68 joints positions were used to form a skeleton of 20 joints relative to locations of articulations of a human body. Only 11 joints were selected for the next steps of feature extraction, excluding arms joints that are not important in the performed gestures. Seven relational features, inspired by Müller [2], representing geometric relationships between joints were also used and are as follows:

- Distance between the right ankle joint and the plane defined by the pelvis, left hip, and left ankle joints.
- Distance between the left ankle and the plane fixed in the right ankle and normal to the vector (right hip, left hip).
- Angle between the vectors (right knee, right hip) and (left knee, right ankle).
- Angle between the vectors (left knee, left hip) and (left knee, left ankle).
- Angle between the vectors (neck, pelvis) and (right hip, right knee). (= angle between right leg and body spine).
- Angle between the vectors (neck, pelvis) and (left hip, left knee). (= angle between Left leg and body spine).
- Angle between the vector (neck, pelvis) and the vector perpendicular to the ground.

Figure 1, taken from [2], illustrates well the main idea of these relational features. The respected features in this figure express whether (i) the right foot lies in front of or behind the body, (ii) the left hand is reaching out the front of the body or not, and (iii) the left hand is raised above neck height or not. We used also 36 features representing distances between each pair of joints of the lower part of the skeleton during an action. In summary, a frame is described by 76 dimensions. The recorded sequences where annotated manually in the four classes (dance steps) cited previously.

3.1. Learning

The learning procedure we follow in our system is illustrated in Figure 2. The extracted features are used to train a left-to-right HMM with no skip transitions for five classes, the four classes cited previously (Maclotte base, Passepied Base, Passepied Fleuret and Back step), in addition to one class representing pauses between performances (neutral pose). A left-to-right model with no skip transitions is a basic model in which the only possible transitions in each frame are either to stay in the same state or to go to the next state. Each model in our system consists of 11 states, and this number was selected empirically as it gives the highest recognition rate.

In addition to the number of states, two probability measures must be defined: transition probabilities $t_{i,j}$ between two states (s_i, s_j) and the probability density functions e_i of the observations in each state s_i. The probability density functions (pdfs) can be either modeled by a mixture of Gaussians or a single Gaussian. In our approach, we have used a single Gaussian as illustrated in Figure 2. The database that served to train our models contains about 8000 frames annotated in 114 steps. Seventy per cent of the database was used for the training phase and the remaining 30% for decoding. We ran 10 different training batches for cross-validation.

3.2. Adaptation

The idea of adaptation is inspired by works performed in the field of speech recognition [18,19] where a speaker-independent system is adapted to improve recognition of a new speaker. This means that, to avoid using a database containing a huge amount of data for every speaker, we can train the system on one or a few speakers data for whom sufficient data are available. For recognition

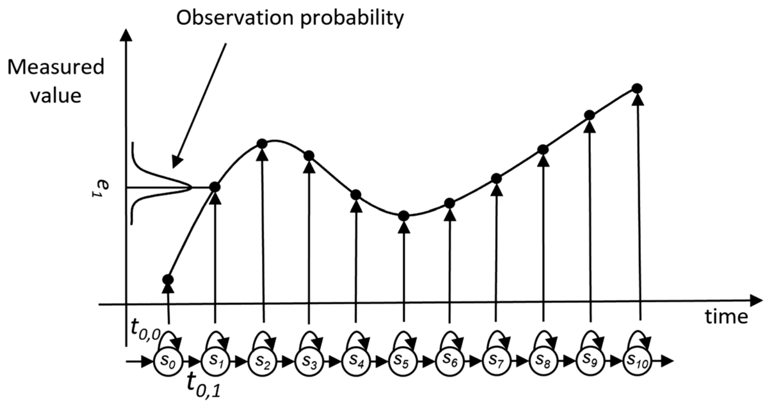


Figure 2. 11-states left-to-right hidden Markov model for learning procedure.

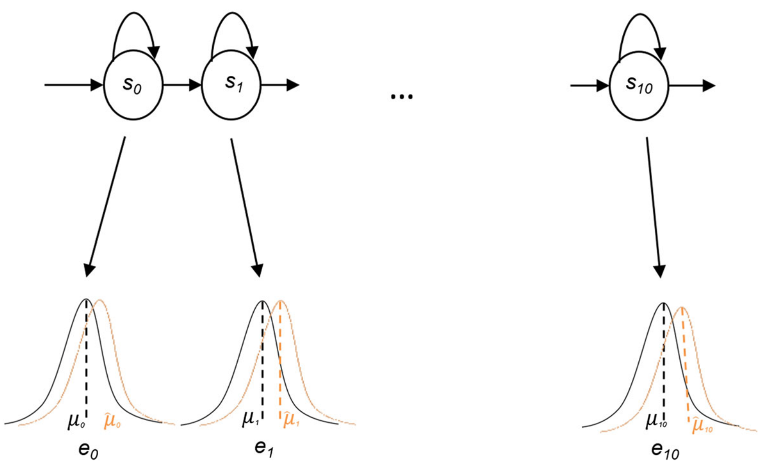


Figure 3. Illustration of the mean only maximum likelihood linear regression procedure.

of a new speaker, few samples from his data can be enough to adapt the models to this speaker and thus create a speaker-dependent speech recognition system. In the present work, we apply a similar approach as we create a "sensor-dependent" gesture recognition and evaluation system based on adaptation procedure [3]. This is performed by using some samples of the same gestures captured by a low-cost sensor that will be used for real-time decoding. This process allows at the same time to have a clean and highly precise data to be analyzed and a system that can recognize gestures from a different sensor.

One of the approach used for adaptation is MLLR. MLLR estimates the parameter of an adapted model by computing a linear transformation of a given speaker-independent models parameter to maximize the observations likelihood. In our system, we have used a *mean only* MLLR method that is already implemented in the HTK toolkit, and in which, a new adapted mean vector $\hat{\mu}$ is calculated. The idea of this method is to shift the mean parameter of the Gaussians in order to have updated models that fit the new data as illustrated in Figure 3. Because only single Gaussians are used to model the pdfs, few samples from the adaptation data can be enough to have an efficient adaptation. These data are captured by the Microsoft Kinect V2 at a framerate of about 30 fps. It contains about 1000 frames (about 17% the size of the training dataset) annotated in the four classes (steps) of the dance.

3.3. Decoding

Decoding is the process of finding the most likely sequence of hidden states corresponding to a new sequence of observation given the parameters of the model. This is performed by using a standard dynamic programming algorithm named Viterbi algorithm.

A major issue when using Viterbi for states decoding is that the sequence must be known in advance, and consequently, it cannot be used for real-time recognition. In our case, this is not important because we want to give the user a final feedback at the end of his or her performance.

3.4. Recognition Results

The first challenge for the recognition process was to select the number of states that gives the highest recognition accuracy and to select the most important features among those cited previously. For this reason, we trained our

DOI: 10.1002/cav

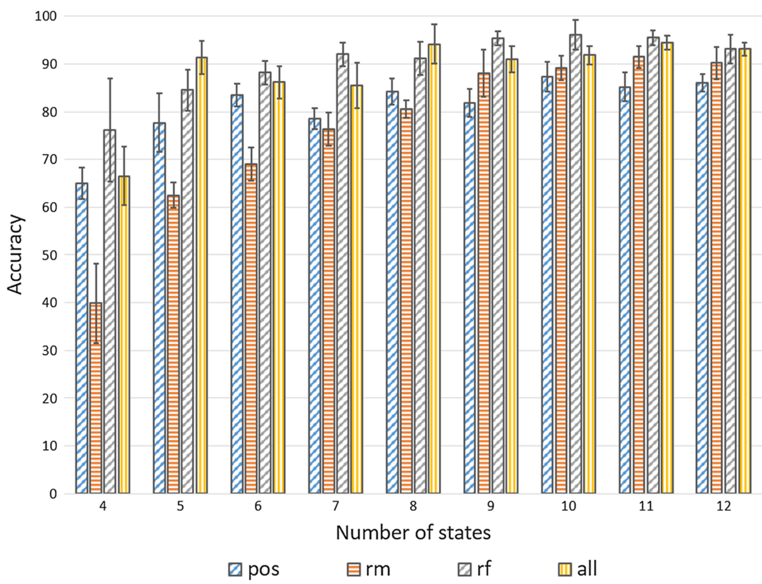


Figure 4. Results of the effect of selected feature sets and number of states on the recognition accuracy.

models on the high precise data by using each time, one single feature category (3D normalized joint positions (pos), relative motions for distances between joint pairs (rm), and the seven relational features (rf)) and then a combination of these features. The number of states was changed between 4 and 12. Ten different training and decoding batches were used for cross-validation. Figure 4 shows the results of accuracy for each case. We observe that in most cases, combinations of features give a higher accuracy than using one category alone except when the number of states equals to 9, 10, or 11; the seven relational features alone gave a higher accuracy than all other feature sets. The highest accuracy value for using relational features alone was 96.17%($\pm$3.07) with a number of states equal to 10. For the combination of feature sets, the accuracy was 94.5%($\pm$1.50) with a number of states equal to 11. We selected these two cases for our next steps.

In fact, several features have been tested, and these three sets are the ones that allowed having high accuracy for both Kinect and Qualisys data. Figure 5 shows the recognition accuracy using only Kinect data for training and test then only Qualisys data for both selected cases. Recognition accuracy using only Kinect data was higher than 87% for both cases, whereas using only Qualisys data, the accuracy was higher than 94%. This shows that the selected features are not easily influenced by the change of motion capture system.

The second challenge of our recognition process was to decode gestures from data captured by a different sensor than the one used for training the original models; in our case, decoding using Kinect V2 data. We applied an adaptation procedure using MLLR, and we compared results of recognition before and after adaptation in the two configurations selected at the previous step and also with. The results are shown in Figure 6. For the first case (relational features—10 states), we do not notice a big difference. The

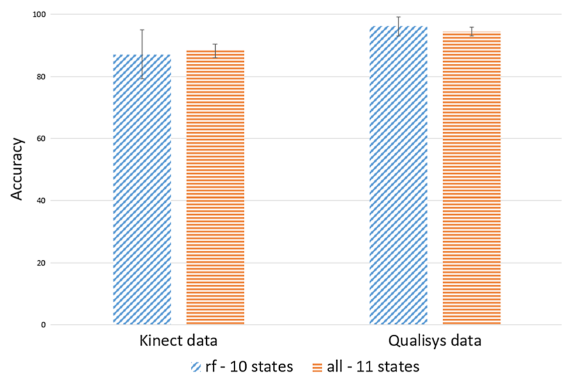


Figure 5. Recognition accuracy using Kinect then Qualisys data for training and test.

recognition accuracy before and after adaptation was about 63.9% for both cases, whereas for the second case (combination of the three feature sets—11 states), the difference is very clear. Before the adaptation, the accuracy was equal to 74.31($\pm$3.03), and after adaptation, it was equal to 81.03($\pm$3.44) . Even with no adaptation, the combination of features gives a higher accuracy than relational features alone. This may be explained by the insufficient number of features for the first case (only seven features), and that these features are not cross-sensors. In other words, they are more affected by the change of sensors. For the next step, we adopted models of the second case where all features are selected and the HMM as a number of states equal to 11.

3.5. Performance Evaluation

For decoding, Viterbi algorithm gives an approximation of the likelihood of the gesture to be recognized given the model. The output is actually a log-likelihood value

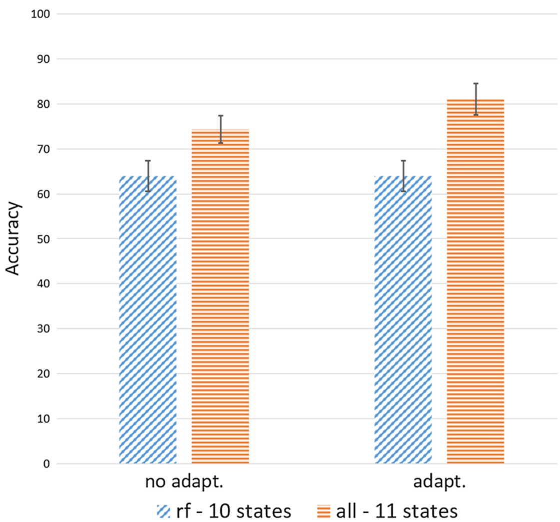


Figure 6. Results of recognition accuracy before and after adaptation for the selected feature sets and hidden Markov model topologies.

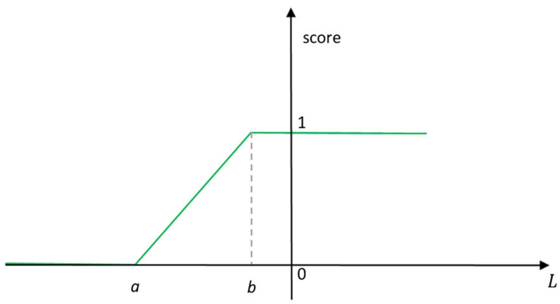


Figure 7. Score function used for mapping the normalized log-likelihood to the score.

that decreases when the length (number of frames) of the sequence increases. We perform time normalization of log-likelihood by dividing by the length of the sequence. The resulting value is not in a limited range and hence it cannot be interpreted by the user. Our goal is to give the user a percentage score comparing his or her performance to a reference represented by the step models. This percentage score can be interpreted by the user as an evaluation of his or her performance whether it is good (higher than 75%), medium (between 50% and 70%), or bad (less than 50%). In order to obtain this score, we map the resulting normalized log-likelihood (L) on the following function:

$$score = \begin{cases} 0, & if\ L < a \\ 1, & if\ L > b \\ \dfrac{(L-a)}{(b-a)}, & \text{otherwise} \end{cases} \tag{1}$$

The function is illustrated in Figure 7. It allows to compute a score that evaluates the performance of the user.

a and b are determined empirically by outputting the values of the log-likelihood when decoding reference gestures. We compared our approach to Kitsikidis one [14] by evaluating an expert and a non-expert of the Walloon dance performing two different styles (Maclotte Base–MB. and Passepied Base–PB.). We evaluate also the performance of the Backward steps (Back.). The results are presented in Table I. The average scores for the expert using our method were 96.06% for Maclotte Base, 99.32% for Passepied Base, and 87.63% for the Backward step, and this is obvious because the models were trained on his data, where using Kitsikidis method, scores were lower: 78.78% for the Maclotte Base step, 94.87% for the Passepied Base step, and 69.41% for the Backward step.

Table I. Comparison between expert and non-expert of Walloon dance performances.

	Method	HMM	Kitsikidis
Expert	MB.	96.06	78.18
	PB.	99.32	94.87
	Back.	87.63	69.41
Non-expert	MB.	87.63	54.73
	PB.	48.48	66.26
	Back.	56.40	68.54

MB., Maclotte Base; PB., Passepied Base; Back., Backward steps; HMM, hidden Markov model.

Table II. Evaluation of the student performance by the expert.

	MB.	PB.	Back.
Expert Eval.	Good	Bad	Medium

MB., Maclotte Base; PB., Passepied Base; Back., Backward steps.

The expert of this dance commented the student's performance and provided an evaluation for each performed gesture. This evaluation is summarized in Table II.

The student performs well the Maclotte step, where the Passepied needs to be improved, and the performance of the Backward step is acceptable. Based on these comments, we can see that our method confirms the expert evaluation. The student had a score of 87.63% for the performance of the Maclotte Base steps and 48.48% for the performance of the Passepied Base step. Kitsikidis method missed the evaluation of the student where the scores estimated from the performance of the two steps (Maclotte Base and Passepied Base) were 54.73% and 66.23%, respectively.

4. APPLICATION

The evaluation system presented in this paper has been used for learning basic steps of traditional Walloon dance, a dance from the south region of Belgium, in a serious game-like environment. This serious game has been developed under the framework of the European project

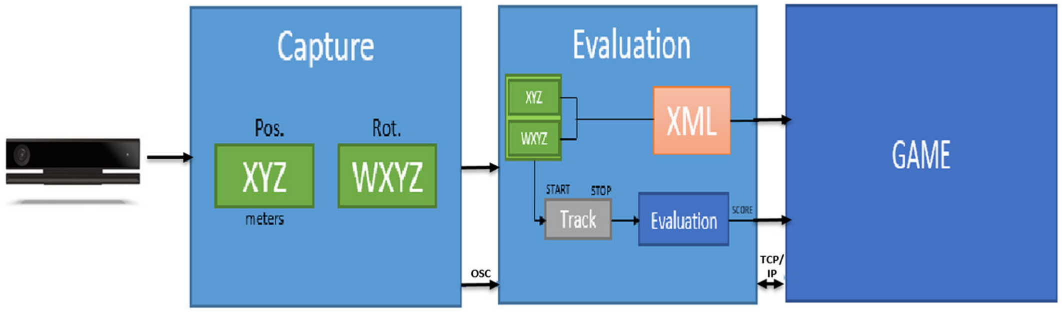


Figure 8. Overall architecture of the communication between MotionMachine and the game.

Figure 9. Overview of the game interface.

i-Treasures.‡ The overall architecture of the module is presented in Figure 8. The Viterbi decoding and the scoring function were implemented within the MotionMachine framework.§ This implementation, which uses the models previously trained by HTK, is the dance step evaluation used within the game.

At runtime, the learner's movements are captured by the Kinect V2 sensor through a module implemented within the MotionMachine framework. These movements are sent to the actual game rendering framework for dance learning, which runs on Unity¶. Figure 9 presents an overview of the Unity game interface. An avatar of the expert is shown in the top right of the figure and the avatar of the learner is shown in the middle of the scene. The learner imitates the expert moves, and then developed module decodes his movements and sends a score value that is displayed on screen. If the learner's performance is quite good (a score $> 50\%$), the game presents the next exercise to be learned, otherwise, the same exercise is presented again and the learner has to try again until he gets the right moves.

5. CONCLUSION AND FUTURE WORKS

In this paper, we have presented an approach for gesture recognition and evaluation based on HMMs. This approach allows on one hand to use a high precise motion capture system to capture reference gestures for better analysis and modeling, and on the other hand, to use a low-cost system for decoding and evaluation, thanks to an adaptation procedure. The system outputs log-likelihood values that are translated into a score to evaluate the learner's gestures. Results showed that the adaptation procedure is important when dealing with different types of data. In addition, the proposed gesture evaluation approach agrees with the expert evaluation. This system has been used to evaluate a traditional dance learner's performance in game-like environment.

In this work, a limited set of gestures from two subjects was used for recognition. Future work will involve increasing the number of gestures, training, and testing with a

‡The i-Treasures project: http://i-treasures.eu/.
§MotionMachine: http://www.numediart.org/motionmachine/.
¶Unity 3D: https://unity3d.com/.

larger number of subjects, testing the algorithm with other kind of gestures than traditional dances and evaluating the game with more learners.

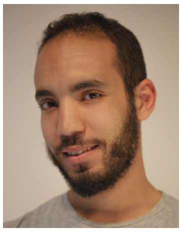
ACKNOWLEDGEMENT

This work has been supported by the European Union (FP7-IC7-2011-9) under grant agreement number 600676 (i-Treasures project).

REFERENCES

1. Mitra S, Acharya T. Gesture recognition: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 2007; **37**(3): 311–324.
2. Müller M. *Information Retrieval for Music and Motion*, Vol. 2. Springer: Berlin, 2007.
3. Laraba S, Tilmanne Joëlle, Dutoit T. Adaptation procedure for hmm-based sensor-dependent gesture recognition. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*. ACM: Paris, France, 2015; 17–22.
4. Ravet T, Tilmanne J, d'Alessandro N. Hidden Markov model based real-time motion recognition and following. In *Proceedings of the 2014 International Workshop on Movement and Computing*. ACM: Paris, France, 2014; 82.
5. Ellis C, Masood SZ, Tappen MF, Laviola JJ, Jr., Sukthankar R. Exploring the trade-off between accuracy and observational latency in action recognition. *International Journal of Computer Vision* 2013; **101**(3): 420–436.
6. Müller M, Röder T, Clausen M. Efficient content-based retrieval of motion capture data. In *ACM Transactions on Graphics (TOG),* Vol. 24*,* ACM, New York, 2005; 677–685.
7. Ofli F, Chaudhry R, Kurillo G, Vidal René, Bajcsy R. Sequence of the most informative joints (smij): a new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation* 2014; **25**(1): 24–38.
8. Blackburn J, Ribeiro E. Human motion recognition using isomap and dynamic time warping. In *Human Motion—Understanding, Modeling, Capture and Animation*. Springer: Berlin, Heidelberg, 2007; 285–298.
9. Vemulapalli R, Arrate F, Chellappa R. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus Convention Center, Columbus, OH, USA, 2014; 588–595.
10. Hong P, Turk M, Huang TS. Gesture modeling and recognition using finite state machines. In *Proceedings.Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, 2000, IEEE; 410–415.
11. de Bettio RW, Silva AHC, Heimfarth T, Freire AP, de Sá AGC. Model and implementation of body movement recognition using support vector machines and finite state machines with cartesian coordinates input for gesture-based interaction. *Journal of Computer Science & Technology* 2013; **13**: 69–75.
12. Bevilacqua F, Zamborlin B, Sypniewski A, Schnell N, Guédy F, Rasamimanana N. Continuous realtime gesture following and recognition. In *Gesture in Embodied Communication and Human-Computer Interaction*. Springer: Berlin Heidelberg, 2009; 73–84.
13. Wu D, Shao L. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus Convention Center, Columbus, OH, USA, 2014; 724–731.
14. Kitsikidis A, Dimitropoulos K, Yilmaz E, Douka S, Grammalidis N. Multi-sensor technology and fuzzy logic for dancers motion analysis and performance evaluation within a 3d virtual environment. In *Universal Access in Human-Computer Interaction. Design and Development Methods for Universal Access*. Springer: Springer International Publishing, 2014; 379–390.
15. Bloom V, Makris D, Argyriou V. G3d: a gaming action dataset and real time action recognition evaluation framework. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE: Providence, Rhode Island, 2012; 7–12.
16. Chan JCP, Leung H, Tang JKT, Komura T. A virtual reality dance training system using motion capture technology. *IEEE Transactions on Learning Technologies* 2011; **4**(2): 187–195.
17. Young S, Kershaw D, Odell J, Ollason D, Valtchev V, Woodland P. *The HTK Book Version 3.0*. Cambridge University Press: Cambridge, 2000.
18. Leggetter CJ, Woodland PC. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech & Language* 1995; **9**(2): 171–185.
19. Leggetter CJ, Woodland PC. Flexible speaker adaptation using maximum likelihood linear regression. In *Proceedings of ARPA Spoken Language Technology Workshop*, Austin, TX, USA, 1995, Citeseer; 110–115.

AUTHORS' BIOGRAPHIES



Sohaib Laraba is a PhD student at the University of Mons (UMONS) in Belgium and an Electronic Engineer from the National Polytehnic School (ENP) of Algiers (Algeria). His research includes analysis, recognition, and assessment of stylistic gestures using different motion capture systems and machine learning techniques.

Joëlle Tilmanne is a postdoctoral researcher at UMONS and is the head of the motion capture and analysis research group at the numediart Institute. She holds a PhD in Applied Sciences from UMONS Faculty of Engineering since 2012 in the field of motion capture data analysis and hidden Markov model-based motion synthesis. She is the co-founder of Hovertone, a young startup active in the domain of creative experience design.