# Hidden Markov Model Based Real-Time Motion Recognition and Following

Thierry Ravet
numediart Institute
University of Mons
Mons, Belgium
thierry.ravet
@umons.ac.be

Joëlle Tilmanne
numediart Institute
University of Mons
Mons, Belgium
joelle.tilmanne
@umons.ac.be

Nicolas d'Alessandro
numediart Institute
University of Mons
Mons, Belgium
nicolas.dalessandro
@umons.ac.be

## ABSTRACT

In this paper, we present a comparison between four HMM-based real-time decoding algorithms for stylistic gait recognition and following. The approach is based on a probabilistic modelling of walking gestures recorded through motion capture. The algorithms are evaluated on their ability to recover the progression of the performed gestures over time in real-time, i.e. as the gestures are performed, and their robustness when the decoding is only performed on a subset of the model dimensions. The performance of studied algorithms are also evaluated in the context of a framework for "gait reconstruction", i.e. where the walking gestures recognised on lower body dimensions are used to synchronously regenerate the upper body dimensions (and vice-versa).

## Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications—*Signal processing*

## Keywords

Motion Capture, Gesture Following, Gesture Recognition, Real-Time Decoding, Hidden Markov Model, Motion Styles

## 1. INTRODUCTION

Nowadays capture, analysis and synthesis of human gestures generate a growing amount of interest from various research groups and industries. The latest developments of motion capture (mocap) technologies have impacted on a wide range of technological solutions, from the expensive high-quality systems targeted for the film industry to the democratised markerless systems based on depth cameras. The wide range of existing mocap technologies can nowadays provide consumers with solutions for gesture acquisition in most situations. Such technologies offer a lot of possibilities for designing Natural User Interaction (NUI).

The motion analysis problem is however less advanced. Analysis techniques based on explicit description of the gestures – and hence on a priori knowledge – exist, but they are generally not robust enough to cope with human motion variability. Furthermore formalising our human understanding and knowledge about human motion is not a trivial task. Machine learning techniques hence seem to be a privileged approach to the analysis and behaviour interpretation of motion sequences. Such techniques consists in "teaching" the machine how to replicate our implicit knowledge about human gestures, with as least explicit description as possible.

In the present work, we address the gesture recognition problem. A usable gesture and action recognition method would ideally be accurate, flexible, easy to extend, real-time, independent from subject identity, robust to occlusions, requiring minimal effort for capturing the data (e.g. markerless and affordable equipment), capable of discriminating between large set of gestures, etc. Since no ideal solution exists, the constraints and mandatory requirements that have to be taken into account in the design of a solution will depend on the considered application and different custom approaches will be built for different problems. To that extend, the efficiency of a gesture/action recogniser tightly rely on how the given problem has been understood and how this understanding has inferred the appropriate choices in the recogniser implementation.

Our approach to gesture recognition is guided by a framework for real-time stylistic gesture mapping. This framework enables the real-time HMM-based recognition of a given gesture sequence from a subset of its dimensions, the covariance-based mapping of the gesture stylistics from this subset onto the remaining dimensions and the real-time synthesis of the remaining dimensions from their corresponding HMMs. This real-time stylistic gesture mapping is beyond the scope of the present paper, but introduces very specific constraints to our recognition task. The foreseen application requires the gesture recogniser to work in real-time, to follow the progression of the recognised gesture frame by frame, to enable the recognition to be performed on any subset of the mocap data dimensions and to take into account the style of the motion implicitly, i.e. without any explicit tagging of the styles achieved by the user in the training database.

The implicit mapping use case also implies that the motion model used for recognition is compliant for synthesis, i.e. the regeneration phase. This is a very strong constraint compared to most existing recognition approaches. This last requirement along with the objective to build a scalable recogniser in which very different gestures can easily be added led us to avoid dimensionality reduction of the mocap data. In this work we propose to use probabilistic approaches and more specifically Hidden Markov Models (HMMs). HMMs have already been used with success in various gesture modelling and analysis related problems [3, 1, 12]. By processing all the dimensions of the full-body mocap data, we model the temporal sequence of multiple channels presenting strong correlations between them. These correlations model the relations that exist between the different body parts during motion and implicitly model the motion stylistic variations. These stylistic variations are the slight variations that occur in realistic gestures for an identical functional pattern (for instance, different styles of walk).

In this paper we present and compare several real-time decoding algorithms based on a partial set of the model dimensions. As a case study, we considered and modelled the full body motion signal corresponding to stylistic gait sequences and compared the recognition results obtained considering motion data dimensions of the whole body, the upper body parts (torso, arms and head) or the lower body parts (pelvis and legs). The paper is organised as follows. Section 2 gives an overview of the previous work related to our problem. Section 3 describes the data that we used and the models that were trained. We explain the four real-time decoding algorithms that were implemented in Section 4. The results of the evaluation are discussed in Section 5. Finally we conclude and prospect future works in Section 6.

## 2. RELATED WORK

Gesture modelling for recognition tasks has been studied with different approaches and some of these techniques show interesting properties for highlighting stylistic components: Principal Component Analysis [15], Hidden Conditional Random Field [16], Conditional Restricted Boltzmann Machine [10] or Dynamic Bayesian Network [7]. Other approaches include Particle Filtering, Deep Neural Network, etc. Exhaustive reviews of literature on gesture recognition can be found for instance in [8, 4].

A lot of authors [1, 3, 17] suggested the use of Hidden Markov Models (HMMs) for the modelling of body motion time series. HMMs integrate directly both the time and the stylistic variability of the motion in their modelling, thanks to their topology. In addition to the recognition applications, HMMs have been more and more widely used for generation, particularly since the development of speech processing tools such as the HTS toolkit (HMM-based speech synthesis system [14]). This further extends the relevance of HMMs as a modelling tool for parameter sequences, as almost the same models can be used to achieve the recognition and generation tasks. Tilmanne and al. have shown recently that HMM-based generation can be very useful for the real-time exploration of a stylistic motion space [11].

Moreover Hueber and al. [6] have explained how to use the Gaussian distributions of the probability density functions to synthesise one target modality according to a correlated source modality. The technique exploits the relation between correlated observation modalities modelled in the HMM. They propose to shift the target models proportionally to the difference between the input data and the statistical parameters of the model. d'Alessandro and al. [5] have demonstrated in a proof-of-concept application that it is possible to generate lower-body movement corresponding to an input upper-body motion during a gait sequence in real-time. They use a forward Viterbi algorithm for decoding the states sequence. In order to improve the recognition module of their system, we compare four different real-time approaches of the Viterbi algorithm in this work[1].

Based on a forward decoding approach, Bevilacqua and al. [1] have proposed a strategy to describe gestures with HMMs based on one single training sequence. They discriminate between gesture models in real-time based on accumulated likelihood and propose a method to decrease the computational load when the number of states in the models is large. Contrarily to this technique, we chose to consider HMMs containing a limited number of states and trained on a large number of training samples in order to model the wide variability of the performed gestures. This approach is explained further in Section 3. Our evaluation is built upon the work of Bloit and Rodet [2]. Their "fusion point" algorithm is a short-term Viterbi that has an interesting property: if you compute the Viterbi decoding on a growing observation sequence, at some point, a given subsequence of past states stabilise and actually correspond to the offline optimal state sequence. The latency introduced by this algorithm can be seen as the minimal latency that leads to the optimal state sequence in a short-term decoding scheme.

## 3. STYLISTIC MOTION MODELLING WITH HIDDEN MARKOV MODELS

The HMM motion model used for gesture recognition was trained using the HTK toolkit (Hidden Markov Model Toolkit [9]). The training datasets were selected in the Mockey database [13] in which an actor was recorded walking while adopting different "styles". The expressive styles selected in this work are the following: proud, decided, sad, catwalk, drunk, cool, afraid, heavy, in a hurry, manly. The motion data were captured at a rate of 30 frames per second (fps). Each frame contains the body skeleton pose data described by 18 joints x 3D angular values = 54 values per frame. The 3D angles were converted into the exponential map parameterisation. The first and second derivatives of the motion data (velocity and acceleration) were also taken into account, and each motion frame is hence described by 162 dimensions. The frames from all the stylistic sequence were annotated into three labeled groups: right step, left step or "garbage" (for the non-walk motion segments). The three class labels correspond to the basic gestures that will be discriminated in the gait model.

We consider one left-to-right four states HMM per "step" gesture to be recognised, plus one model for "garbage" motion (sometimes called the "filler" model). However, for continuous gesture recognition, an overall model of the motion is required. In order to obtain such a model, the basic gesture left-to-right HMMs and the filler model need to be connected. We chose not to take any a priori assumption on the possible sequence of gestures, and hence to connect all the basic HMMs in parallel, which means that each gesture can follow any other gesture. This overall model is illustrated for the walk motion use case in Figure 1. The overall model hence consists in nine states for the walk step recognition example: four states per step, plus one state for the filler or garbage model.

[1]An illustration of gait reconstruction for upper body part can be found at `http://youtu.be/g_CqEd_joV4`

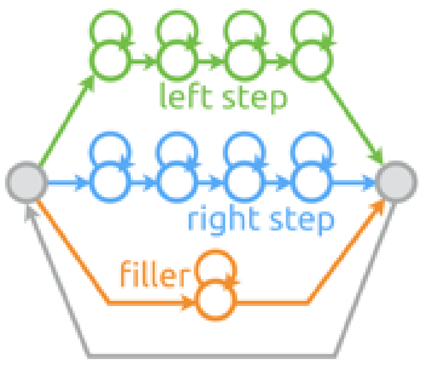

**Figure 1: Overall motion model of walk combining the basic gestures HMMs – left and right steps – plus the garbage/filler model.**

The 30fps mocap walk database that served to train the left-to-right step models contains 12000 frames segmented and annotated in 500 steps. The left and right step models were trained using 80% of the database walk sequences, which were randomly selected. The decoding quality was tested on the remaining 20% of the walk sequences. We ran 10 different training and evaluation batches. The garbage model was trained with a collection of non-gait mocap sequences containing 66000 frames.

## 4. REAL-TIME RECOGNITION

When using HMMs, the recognition problem consists in decoding the most likely sequence of hidden states corresponding to a new sequence of observations, given the parameters of the model. Since each state is linked to one of the three labels (each label corresponding to one basic gesture HMM), decoding the most likely sequence of states also decodes the most likely sequence of labels. A state-by-state decoding enables not only to recognise the gesture (i.e. the label) but also to follow the progression of the gesture execution on a frame-by-frame basis. This state decoding is performed by using the Viterbi algorithm, a standard dynamic programming algorithm.

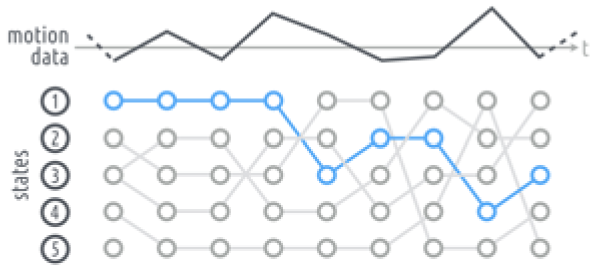

**Figure 2: HMM-based motion recognition: illustration of the state decoding problem using Viterbi.**

One major issue with the use of the standard Viterbi algorithm is that the whole sequence of observations needs to be known in advance (Figure 2). It can hence not be used as such for real-time gesture recognition. This is why we had to implement and test several adaptations of the Viterbi algorithm to real-time motion recognition. In this work, four different approaches for online gesture recognition have been implemented and tested: forward only, sliding window, state stability and Bloit and Rodet's fusion point.

### 4.1 Forward only

The forward only algorithm is quite straightforward: the probability of being in each state at each time $t$ is computed in the same way as for the standard Viterbi algorithm. However, at each time $t$, a decision is taken and the most likely state is considered as the decoded state. The path is hence defined time stamp per time stamp (or in other words frame by frame), and impossible state transitions might be found in the decoded path since a decision is taken based on the most likely state regardless of the existence of possible transitions from the past decoded state (Figure 3).

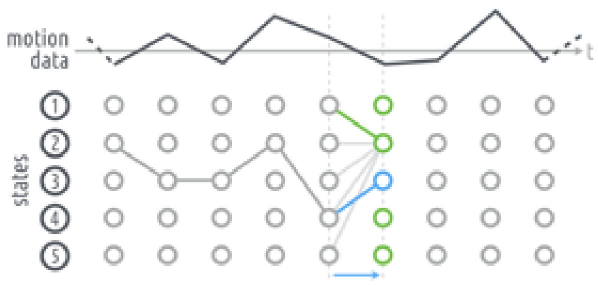

**Figure 3: Real-time Viterbi decoding: the forward only approach, i.e. decision taken at each state.**

### 4.2 Sliding window

The sliding window implementation is illustrated in Figure 4. It consists in computing the standard Viterbi path, but on a window of fixed number of observations/states, as displayed in green in Figure 4. Once the standard Viterbi path has been decoded for that time window, the first state of the path is considered as decoded (as illustrated in blue in the Figure 4). Once the decision is taken, the window slides one observation/state in the future and the same procedure is repeated. This procedure is more accurate than the forward only implementation, but introduces a small delay since the decision is taken after a duration corresponding to the time window. Another typical problem is to determine the optimal length of that window.

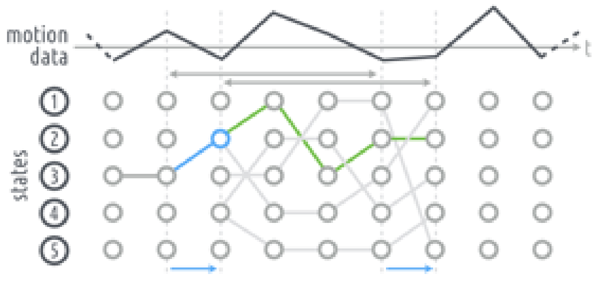

**Figure 4: Real-time Viterbi decoding: the sliding window approach, i.e. decoding on a subpath and validation of the oldest state as decoded.**

## 4.3 State stability

The state stability algorithm consists in computing the Viterbi path following the standard method, starting from the most likely state, but on consecutive increasing-duration time windows (Figure 5). Once the oldest state of the different Viterbi paths converge to the same state for a given minimal amount of times, this state is considered as the most likely state, and the same procedure begins again, starting with this newly decoded state as the first state of the Viterbi paths to be decoded. A variable length delay is introduced with this algorithm. A maximum delay is also defined so that if there is no convergence after a given amount of iterations, a suboptimal decision is taken.

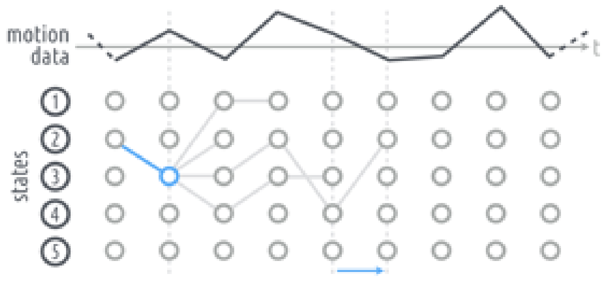

**Figure 5: Real-time Viterbi decoding: the state stability approach, i.e. monitoring of past states variations and validation upon stabilisation.**

## 4.4 Fusion point

The fusion point algorithm has been described by Bloit and Rodet [2] and is illustrated in Figure 6. Its result is equivalent to the offline Viterbi decoding. The standard Viterbi algorithm is computed on a time window of fixed length. That length is increased by one sample until all paths computed in the forward Viterbi procedure converge to a common sequence of states, as illustrated in blue in Figure 6. These states are then considered as decoded, and the time window is shifted so as to begin with the last decoded state. Once again, such an approach introduces a variable size delay in the decoding process. As for the state stability algorithm, a limit (maximum delay) is set so that if there is no convergence after a fixed number of iterations, a decision is taken. In order to limit the delay, as explained by Bloit and Rodet, it is necessary that all the states are connex: any state must be reachable from any other state. In order to ensure this condition, the overall motion model structure illustrated in Figure 1 had to be modified modified by adding an epsilon transition probability between all the states that were previously not connected.

## 5. RESULTS

The performances of the decoding algorithms described in Section 4 were evaluated on ten walk sequences from the Mockey database (one from each walk style) that were not used in the training set (cross-validation). Ten different training and evaluation batches were performed.

The reference state sequence which was used for the evaluation of the state decoding precision is a decoding of the full body motion data with an offline Viterbi algorithm that provides the optimal solution. We conducted the evaluation on three different subsets of the data dimensions. In the first one we considered all the full body joints data as input. In the second one only the upper-body joints data (7 nodes) was considered for recognition. In the third one we only considered the motion data from the lower-body joints (12 nodes). We used the hips joint motion information for both the upper- and the lower-body sets of data.

The size of the sliding window in the sliding window Viterbi algorithm was tuned to six frames by studying the results of the fusion point analysis on the whole set of test signals. These three different subsets enable us to compare the abilities and the robustness of each algorithm to decode the gait motion signals with all or just a part of the captured data. Table 1 shows the results of model decoding between the three class labels (left step, right step and garbage) in comparison with the hand labeled solution, which corresponds to an evaluation of the gesture recognition. Table 2 illustrates the rate of states decoded similarly between the reference setup (offline Viterbi algorithm considering all body joints) and each other configuration, which corresponds to an evaluation of the gesture recognition and following.

If we just consider the discrimination between left step and right step, the problem itself is not complex. So we do not observe any significant improvement with other approaches compared to the "forward only" algorithm that still provides good performances, as illustrated in Table 1. However the gesture following required for instance in the mapping use case is very sensitive to the quality of state decoding. Here the "forward only" algorithm is clearly outperformed by the other ones as we can see in Table 2 and in Figure 7. The

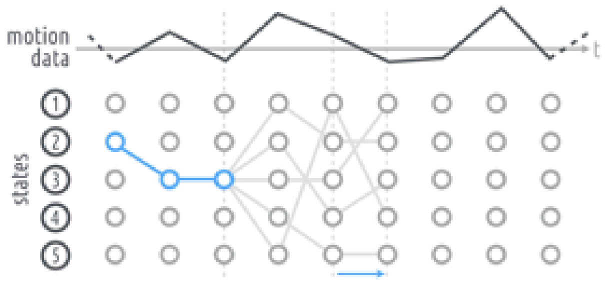

**Figure 6: Real-time Viterbi decoding: the fusion point approach as described by Bloit and Rodet [2].**

**Table 1: Model identification precision in comparison with the hand labeled solution (mean ± SD).**

| | Body joints subgroup | | |
|---|---|---|---|
| Algorithm | all | Lower part | Upper part |
| Offline | 0.9449 ±0.0087 | 0.9308 ±0.0136 | 0.7717 ±0.0243 |
| Forward only | 0.9374 ±0.0102 | 0.9313 ±0.0140 | 0.7604 ± 0.0274 |
| Sliding window | 0.9448 ± 0.0085 | 0.9298 ±0.0137 | 0.7796 ±0.0272 |
| State stability | 0.9453 ±0.0085 | 0.9289 ±0.0141 | 0.7771 ±0.0278 |
| Fusion point | 0.9446 ±0.0087 | 0.9283 ±0.0140 | 0.7739 ±0.0287 |

**Table 2: State decoding precision in comparison with the reference solution (mean ± SD).**

| | Body joints subgroup | | |
|---|---|---|---|
| Algorithm | all | Lower part | Upper part |
| Offline | 1.0000 ±0.0000 | 0.7014 ±0.0480 | 0.5762 ±0.0463 |
| Forward only | 0.8777 ±0.0098 | 0.6710 ±0.0494 | 0.5241 ±0.0507 |
| Sliding window | 0.9969 ±0.0021 | 0.7019 ±0.0493 | 0.5760 ±0.0543 |
| State stability | 0.9789 ±0.0050 | 0.7007 ±0.0501 | 0.5727 ±0.0546 |
| Fusion points | 0.9715 ±0.0060 | 0.7012 ±0.0481 | 0.5783 ±0.0528 |

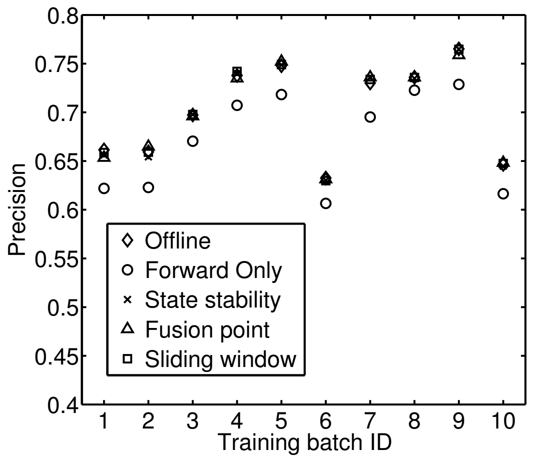

**Figure 7: Dispersion of the state decoding precision for models trained with 10 different data batches when considering the motion data from the lower-body joints only.**

differences between the optimal solution and the fusion point solution are caused by the modification that we have done in the transition matrix for that last case.

In the mapping use case, the smoothness of the synthesised movement is insured by respecting the left-to-right structure of the HMM model. The offline Viterbi algorithm forbids the state-to-state transitions that do not fit the overall HMM as illustrated in Figure1. However the solutions decoded by approximate real-time algorithms may transgress this rule. Table 3 shows the computed ratio between the number of these transgressing transitions and the number of decoded states for every evaluation batch. Although the worst case occurs with the forward only algorithm, we can also observe a significant difference between the other solutions: the state stability algorithm provides less good performance than the other two. This may generate an increasing number of discontinuities in the rendering of the mapping use case.

In Figure 8 we show the difference between the four algorithms for a gait sequence decoding on the full-body capture in reference to the offline Viterbi reference solution. The indices 1 to 4 correspond to the right step model, 4 to 8 to the left step model and 9 to the garbage model.

**Table 3: Ratio between the number of transitions transgressing the left-to-right HMMs structure and the number of decoded states (mean ± SD).**

| | Body joints subgroup | | |
|---|---|---|---|
| Algorithm | all | Lower part | Upper part |
| Offline | 0.0000 ±0.0000 | 0.0000 ±0.0000 | 0.0000 ±0.0000 |
| Forward only | 0.0494 ±0.0043 | 0.0356 ±0.0044 | 0.0516 ±0.0041 |
| Sliding window | 0.0009 ±0.0004 | 0.0015 ±0.0007 | 0.0162 ±0.0042 |
| State stability | 0.0207 ±0.0041 | 0.0074 ±0.0022 | 0.0299 ±0.0033 |
| Fusion points | 0.0155 ±0.0046 | 0.0028 ±0.0011 | 0.0209 ±0.0053 |

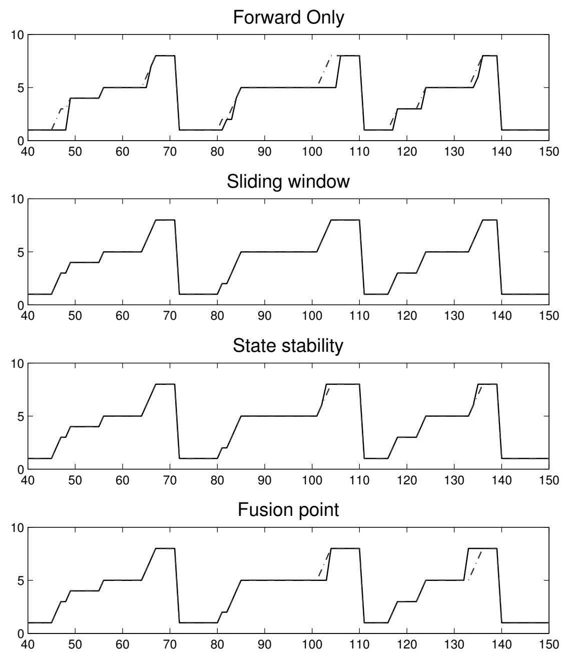

**Figure 8: Viterbi decoding: comparison of four different approaches (dotted line: offline Viterbi solution used as the reference point for comparison).**

## 6. CONCLUSION AND FUTURE WORK

In this paper we have presented four different approaches for real-time gesture recognition and following based on adaptations of the standard Viterbi algorithm. These four implementations have been evaluated and tested on a very simple proof-of-concept walk reconstruction use case. The results show that running the fusion point algorithm to determine the window size and then running the sliding window algorithm with such window size provides a solution that ensures a very good accuracy of state decoding compared to a forward only approach, while introducing only a reasonable six-frame delay (with a frame rate of 30). The short-term Viterbi based on the fusion point decodes the optimal state sequence by chunks. The average length of these successively-decoded chunks is a good image of the past gestural information that is required by the Viterbi algorithm to reach its optimal decoding behaviour. Therefore this average length can be seen as the length of the buffer of decoded states that a given system needs to operate Viterbi-based recognition correctly for a given set of input gestures.

The evaluation was performed on sample walk sequences selected in the Mockey database. However, although the evaluation sequences were not used for training the model, they still come from the same database as the training data, and were hence recorded with the same actor, in the same settings, etc. We plan to extend this evaluation in the future to other mocap sequences recorded with different setups (other actors, other mocap system, etc.) in order to further validate our approach and to improve its robustness. A further step will be to study the necessary adaptation to permit the use of the models trained with accurate and precise motion capture database for recognition of observation sequences obtained with low-cost but less performing and less accurate motion capture systems. This is a challenging problem since data generated by popular instrumentation tools like the Kinect (noisy skeleton poses in Cartesian coordinates) require a consequence amount of preprocessing and cleaning in order to enable a frame by frame gesture recognition and following precise enough to drive applications such as the gait reconstruction application for which the recognition system presented in this paper was designed.

The use case tested the gesture recognition on a very limited set of gestures, from one single subject. Future work will involve increasing the number of gestures in the recogniser vocabulary, training the model and testing the system with a larger number of subjects.

## 7. ACKNOWLEDGMENTS

T. Ravet and N. d'Alessandro are funded by regional funds respectively called Region Wallonne GREENTIC (convention number 1317957) and Region Wallonne FIRST Spin-Off. J. Tilmanne is supported by the European Union, 7th Framework Programme (FP7-ICT-2011-9), under grant agreement no 600676 (i-Treasures project).

## 8. REFERENCES

[1] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana. Continuous realtime gesture following and recognition. *Gesture in embodied communication and human-computer interaction*, pages 73–84, 2010.

[2] J. Bloit and X. Rodet. Short-time viterbi for online hmm decoding: Evaluation on a real-time phone recognition task. In *ICASSP*, pages 2121–2124. IEEE, 2008.

[3] M. Brand and A. Hertzmann. Style machines. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 183–192, 2000.

[4] B. Caramiaux and A. Tanaka. Machine Learning of Musical Gestures. In *Proceedings of the 13th Conference on New Interfaces for Musical Expression (NIME'13)*, 2013.

[5] N. d'Alessandro, J. Tilmanne, M. Astrinaki, T. Hueber, R. Dall, T. Ravet, A. Moinet, H. Cakmak, O. Babacan, A. Barbulescu, V. Parfait, V. Huguenin, E. Kalaycı, and Q. Hu. Reactive Statistical Mapping: Towards the Sketching of Performative Control with Data. *IFIP Advances in Information and Communication Technology (AICT)*, page to appear, 2014.

[6] T. Hueber, G. Bailly, and B. Denby. Continuous Articulatory-to-Acoustic Mapping using Phone-Based Trajectory HMM for a Silent Speech Interface. In *Proceedings of Interspeech, ISCA*, 2012.

[7] M. Lau, Z. Bar-Joseph, and J. Kuffner. Modeling spatial and temporal variation in motion data. In *ACM SIGGRAPH Asia 2009 papers*, pages 171:1–171:10, New York, NY, USA, 2009. ACM.

[8] S. Mitra and T. Acharya. Gesture Recognition: A Survey. In *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, pages 311–324, 2007.

[9] U. of Cambridge. The hidden markov model toolkit (htk). `http://htk.eng.cam.ac.uk`, 2009.

[10] G. W. Taylor and G. E. Hinton. Factored Conditional Restricted Boltzmann Machines for Modeling Motion Style. In *Proc. 26th International Conference on Machine Learning*, pages pp 1025–1032, 2009.

[11] J. Tilmanne, N. d'Alessandro, M. Astrinaki, and T. Ravet. Exploration of a Stylistic Motion Space Through Realtime Synthesis. In *Proceedings of the 8th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP'14)*, 2014.

[12] J. Tilmanne, A. Moinet, and T. Dutoit. Stylistic gait synthesis based on hidden Markov models. *Eurasip journal on Advances in Signal Processing*, 2012:72(1):1–14, 2012.

[13] J. Tilmanne and T. Ravet. The mockey database. `http://tcts.fpms.ac.be/~tilmanne/`, 2010.

[14] Tokuda et al. HMM-Based Speech Synthesis System (HTS). `http://hts.sp.nitech.ac.jp`, 2008.

[15] N. F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision*, 2(5):371–387, 2002.

[16] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1521–1527, 2006.

[17] Y. Wang, L. Xie, Z.-Q. Liu, and L.-Z. Zhou. The somn-hmm model and its application to automatic synthesis of 3d character animations. In *SMC*, pages 4948–4952. IEEE, 2006.