Exploration of a Stylistic Motion Space Through Realtime Synthesis

Joëlle Tilmanne, Nicolas d'Alessandro, Maria Astrinaki and Thierry Ravet
numediart, Institute for Creative Technologies, University of Mons, Mons, Belgium

Keywords: Motion Capture, Motion Style, Realtime Synthesis, Hidden Markov Models.

Abstract: We present a first implementation of a framework for the exploration of stylistic variations in intangible heritage, recorded through motion capture techniques. Our approach is based on a statistical modelling of the phenomenon, which is then presented to the user through a reactive stylistic synthesis, visualised in real-time on a virtual character. This approach enables an interactive exploration of the stylistic space. In this paper, a first implementation of the framework is presented with a proof-of-concept application enabling the intuitive and interactive stylistic exploration of an expressive gait space.

1 INTRODUCTION

Preservation of patrimony, and more especially of intangible cultural heritage (ICH), could gain a completely new dimension thanks to ICT which have appeared and matured in the last decades. Human expert gesture is the essence of most expressions of ICH: dance performances, craftsmanship, music performances, etc. The development of motion capture (mocap) technologies, becoming more precise, less invasive and more affordable, has hence logically set it as an unavoidable approach to intangible patrimony preservation. Mocap enables the transformation of tridimensional human movements into a digital form. This transformation is made through the approximation of the complex human skeleton and body by a simplified kinematic chain of body segments or a reduced set of body joints. Different technologies coexist, each one with its advantages and drawbacks, but solutions can be found for most problems, and it becomes a common tool for performance capture.

However patrimony preservation should aim at preserving a global *know-how*, not one random occurrence or expression of the performance, nor a restricted subset of the overall patrimony. Several representative performances should hence be recorded, ideally from different performers, or expressing different styles if applicable. Unfortunately this global know-how, which can also be seen as a the intangible heritage which should be preserved, is hard to capture and to present in a meaningful way. Three main limitations arise with the use of mocap technologies for capturing expert gestures in performances.

The first one appears when aiming at preserving the overall know-how through several recordings, from several performers, with different styles, as the amount of mocap data can rapidly become quite huge. Inter and intra subject variabilities, stylistic factors and external factors influence each recording, and the dimensionality of mocap data itself is high. The amount and high variability of the recorded data makes it hard to analyse and to use.

A second limitation comes from the lack of interactivity when playing back prerecorded motions. Mocap data is a recording of a performance. There is hence no interactivity when viewing it again. If a high number of performances have been recorded, browsing the motion database can be quite difficult and not intuitive, as the user gets lost in the contents.

The third major issue with mocap is the representativity of the data itself. Cartesian coordinates of body joints or angles between body segments are concepts which are difficult to apprehend and do not match our human experience and expertise of body motions. The difficulty to interpret the evolution of tridimensional angles or coordinates over time make it necessary to visualise mocap data through their projection on 3D virtual characters.

The approach presented in this paper aims at counterbalancing these drawbacks by proposing an alternative tool for visualising and understanding high-dimension mocap databases that exhibit deliberate stylistic variations. Our stylistic exploration tool is based on statistical models which can be considered as a summary of the mocap data, and are used for realtime synthesis of motions with reactive stylistic

Tilmanne J., d'Alessandro N., Astrinaki M. and Ravet T..
Exploration of a Stylistic Motion Space Through Realtime Synthesis.
DOI: 10.5220/0004876008030809
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (IAMICH-2014), pages 803-809
ISBN: 978-989-758-004-8

control. The visualisation of the data is presented through projection of the synthesised motion curves onto a virtual character. This paper presents a proof-of-concept of this motion style exploration tool applied to the case study of walk. In Section 2, we give an overview of the statistical modelling that we used in this research. Section 3 focuses on the realtime and reactive parameter generation strategy, while Section 4 presents the overall application design. Then some discussion about the results is given in Section 5 and Section 6 concludes.

2 STATISTICAL MODELLING OF MOTION CAPTURE DATA

2.1 Motion Database

The case study presented in this paper explores a stylistic space of human gait. The models were trained thanks to the Mockey stylistic walk database (Tilmanne and Ravet, 2010), which aims at studying the expressivity of walk motions. In this database, a single actor was recorded walking back and forth while adopting eleven different "styles". These styles corresponded to different emotions, morphology personifications, or situations, and were arbitrarily chosen because of their recognizable influence on walk, as illustrated in Figure 1. The acted styles were the following: proud, decided, sad, cat-walk, drunk, cool, afraid, tiptoeing, heavy, in a hurry, manly.

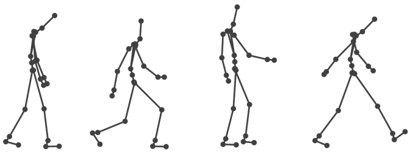

Figure 1: Four example postures from the Mockey database. From left to right: sad, afraid, drunk and decided walks.

The motion was captured thanks to an inertial mocap suit, the IGS-190 from Animazoo (Animazoo, 2008), containing eighteen inertial sensors. The motion data is described by the evolution over time of the 3D Cartesian coordinates of the root of the skeleton, along with the eighteen 3D angles corresponding to the orientation of the skeleton root and of the seventeen joints of the simplified skeleton used to represent the human body (Figure 1). The global position of the skeleton was discarded in our application, since it is extrapolated in the Animazoo system based on the angle values, and can be recalculated in the same way after synthesis. Each body pose is hence described by $18 * 3D = 54$ values per frame. We chose to model the rotations of the eighteen captured joints rather than the 3D Cartesian coordinates of these joints in order to ensure that the fixed limb length constraints were respected in the synthesised motion: as only rotations are applied to the fixed limb length skeleton definition, there will be no length deformation of the limbs after synthesis. We converted the 3D angles from their original Euler parameterisation to the exponential map parameterisation (Grassia, 1998), which is locally linear and where singularities can be avoided. The motion data was captured at a rate of 30 fps. The walk sequence were annotated into right and left steps, thanks to an automatic segmentation algorithm based on the hips joint angles. These two class labels correspond to the basic motions which will be represented in our walk model.

2.2 Hidden Markov Models

Hidden Markov Models (HMMs) are widely used for the modelling of time series, and have been used for motion modelling and recognition since the nineties. One of the advantages of using HMMs is that they exempt from using time warping, needed in most approaches in order to align sequences prior to analyse them or extract the style component among them. HMMs integrate directly in their modelling both the time and the stylistic variability of the motion, thanks to their statistical nature. In addition to the recognition applications, the last decade has seen a rising interest for the use of HMMs for generation, especially with the development of tools such as the HTS Toolkit developed for speech synthesis (Tokuda et al., 2008).

A HMM consists of a finite set of states, with transitions between the states governed by a set of probabilistic distributions called transition probabilities. Each state is associated with an outcome (more generally called observation) probability distribution. Only this observation is visible, the state is called *hidden* or *latent*: at each time t, the external observer sees one observation o_t, but does not know which state produced it. HMMs are double stochastic processes, since both the state transitions and the output distributions are modelled by probabilistic distributions.

Particular HMM structures can be designed for specific applications. The basic left-to-right HMM with no skip transitions illustrated in left side of Figure 2 is an example of such a specific HMM which will be used in our motion modelling and synthesis application. A left-to-right model with no skips is a model in which the only possible state transitions at each time are either to stay in the same state or to go to the next state. The complete characterisation of a HMM requires the specification of the number of states, and the definition of two probability measures: the state transition probabilities $t_{i,j}$ between each states pair (s_i, s_j) and the probability density functions (pdfs) e_i of the observations in each state s_i. In continuous HMMs, these pdfs are most often modelled by a mixture of Gaussians, or single Gaussians as illustrated in Figure 2. Following the approach proposed by (Yoshimura et al., 1998) for HMM-based generation, the transition probabilities can be replaced by Gaussian state duration pdfs d_i, as illustrated in the right side of Figure 2. The HMM is then called a *Semi Hidden Markov Model* or *SHMM*. A compact notation is often used to refer to all the parameters defining the HMM or SHMM, and such a set of the model parameters is written λ.

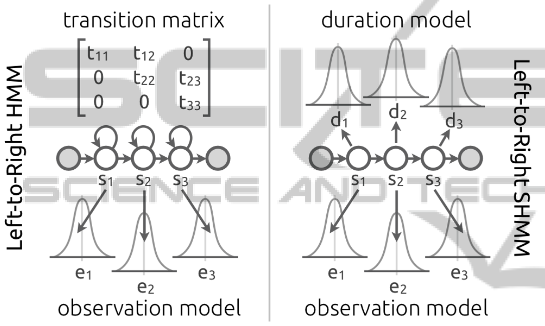

Figure 2: A simple three-states left-to-right HMM with no skip with its associated observation pdfs. A classical HMM is presented on the left side, while its SHMM correspondence is presented on the right side.

2.3 HMM-based Motion Modelling

HTS is the framework we used for training our models and for synthesis. It was designed specifically for speech synthesis. However very strong similarities can be found in the speech and motion modalities. The left-to-right structure of phonemes corresponds to the structure of basic motions, identity, emotions and physical characteristics will influence both speech and motion modalities, etc. HMM-based motion modelling and synthesis can hence greatly benefit from tools developed primarily for speech.

In previous work (Tilmanne et al., 2012), we transposed the HTS speech designed training and synthesis procedure to our motion use case. In the present work, walk is modelled using only two SHMMs, one for the right step and one for the left step. We modelled each step with a five-states SHMM, with multivariate diagonal Gaussian observation pdfs and univariate Gaussian duration pdfs. For reasons related to the parameter generation (see Section 3), the first and second derivatives of the motion data are also modelled. Our walk model consists in 2 (one SHMM per step) * 5 (states) * [2 (mean and variance of duration pdfs) + 2 (mean and variance of observation pdfs) * 54 (dimensionality of mocap data) * 3 (static, first and second derivatives)] = 3260 parameters.

In a first stage, a global walk model is trained using the complete Mockey database. For each one of the eleven walk styles, this generic walk model then undergoes an adaptive training with a reduced set of data corresponding to the target style only. After adaptation, we obtain twelve distinct stylistic walk models: one neutral style model and one model for each one of the eleven Mockey styles. Thanks to the adaptive training, all models are adapted from the same basis model and are therefore aligned.

3 GENERATION OF MOTION PARAMETERS BASED ON "TRAJECTORY HMMS"

Although HMMs have already been tried for character animation (Brand and Hertzmann, 2000; Li et al., 2002; Wang et al., 2006), our prototype relies on a different approach towards HMM-based parameter generation. Indeed, as described in Section 2, the core concept behind our motion modelling framework is the adaptation of advanced HMM-based speech processing algorithms to the motion use case. Over the last decade, the speech research community has come up with very innovative solutions involving HMMs. Among these innovations, Tokuda *et al.* have proposed an algorithm for the use of SHMMs in speech synthesis (Tokuda et al., 2000). This idea has brought a new category of *statistical parametric* speech synthesizers and the HMM-based speech synthesis toolkit, called HTS (Tokuda et al., 2008), has become a reference in the field. In Section 3.1, we describe the algorithm that turns SHMMs into *trajectory HMMs*, enabling the synthesis of realistic parameter trajectories, and we validate this approach for our motion models. Our motion exploration application relies on two other innovations from the HTS research field. The first one turns trajectory HMMs into *reactive HMMs*, i.e. achieves the parameter generation in realtime, and is explained in Section 3.2. The second one is called *model interpolation* and Section 3.3 shows how we use it to create a stylistic motion space.

3.1 Maximum Likelihood Parameter Generation: Trajectory HMMs

The relevance of the HTS algorithm in synthesising high-quality speech trajectories relies on a core algorithm by Tokuda *et al.* called the *Maximum Likelihood Parameter Generation* or *MLPG* (Tokuda et al., 2000). Practically it means that we try to generate new parameter trajectories that are the *most likely* to be observed, for a given statistical model λ.

As described in Section 2, each class label of the database actually corresponds to a N-state left-to-right SHMM. When synthesising a new sequence, a transcription of the sequence of class labels to be synthesised must be available, and each class label *queries* its corresponding model. In our gait modelling use case, $N = 5$, although all the Figures of this Section display 3 states s_i for readability reasons. For each state s_i, the SHMM contains a univariate Gaussian distribution d_i as the state duration model and a multivariate Gaussian distribution e_i as the observation vector model. The process described below works for any series of class labels (not just one), as the SHMMs can be concatenated together before the parameter generation. However in our walk synthesis use case, the walk sequence will always correspond to a succession of right and left steps, and the walk sequence model will hence consist in a concatenation of five-states models corresponding alternatively to the right and left step models.

3.1.1 "Unwrapping" the Duration Models

The first phase in generating a parameter trajectory from the queried SHMM is to *solve* the duration model for each state. According to the Maximum Likelihood criterion, each state s_i of the SHMM must last the most likely amount of time, i.e. the mean of its corresponding duration probability density function d_i. Therefore the so-called *graphical model* of the SHMM is constructed by repeating each s_i a given amount of times, corresponding to $E(d_i)$, where E means expectation, as described in Figure 3.

This approach advantageously replaces the typical transition matrix of HMMs with an explicit model, making this first "unwrapping" phase much easier to achieve, but also to control. Indeed the duration of each state is very visible and accessible. For instance, the five states used in our modelling of gait are expected to be different phases of the steps. Therefore we could easily influence the motion stylistics by shortening or lengthening some of the state durations.

3.1.2 Generation of Parameter Trajectories from the Observation Models

Once we have built the graphical model from duration models, we have a left-to-right structure that supports the evolution of observation statistics over time. Practically it means that we can associate emission probabilities to any point on the timeline of the expected parameter trajectory. Figure 4 (top part) gives an overview of this alignment process, as the following step of what has been created in Figure 3.

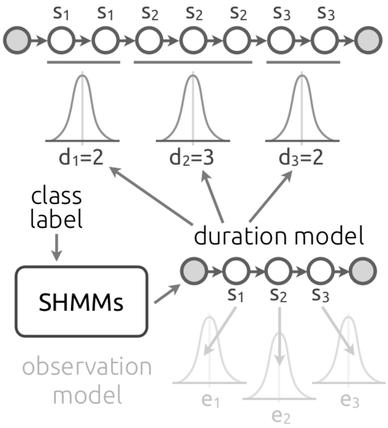

Figure 3: Illustration of the overall process of solving the duration models contained in a given SHMM (queried by the class label). For each state s_i, the mean of each duration pdf d_i is applied on the amount of repetitions of the state, in order to build the SHMM graphical model.

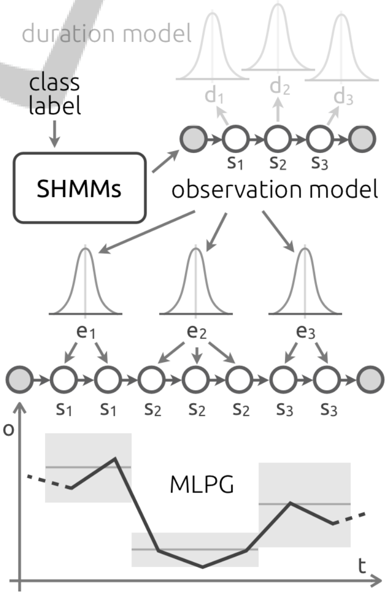

Figure 4: Illustration of the overall process of parameter generation from observation vector statistics, as aligned on the graphical model. The MLPG adds operations to this process so that it generates a smooth, most likely trajectory.

However solving this graphical model under the strict Maximum Likelihood criterion would result in emitting the mean of the observation vector for each state s_i. The resulting trajectory would be a step function with abrupt transitions when i is incremented, e.g. when s_2 switches to s_3. The MLPG algorithm solves this issue by adding two more operations:

- Before computing observation statistics, observation vectors o are extended with their first and second derivatives, so that $O = [\, o \, \Delta o \, \Delta\Delta o \,]$. Then the extended observation vectors O are used instead of o. Such HMMs extended with some of the observation derivatives are called *trajectory HMMs*.
- A specific matrix operation (Tokuda et al., 2000) is achieved on the O vectors and parameters of the statistical models λ so that the Maximum Likelihood is found on the whole sequence, instead of state by state. As this process is achieved on O and not o, the likelihoods of the first and second derivatives are also optimised, resulting in smooth o trajectories. The smoothing process is illustrated in Figure 5 (bottom part).

3.1.3 Realistic Motion Synthesis

As a result of applying the MLPG algorithm on motion vectors, we are able to generate very realistic motion parameters from the class labels. In our use case, we have used either class labels for one average gait model or differentiated class labels for each style. In both cases, we have verified with subjective tests that the animated characters were walking in a very natural manner (Tilmanne et al., 2012).

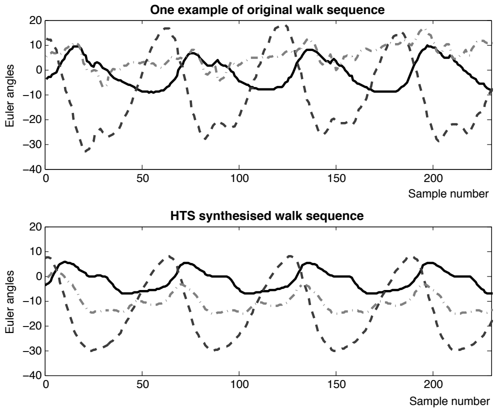

Figure 5: Comparison of original gait motion trajectories vs. the ones synthesised with the MLPG algorithm. The plot represents the evolution of the 3D Euler angle of the left hip during a 7.7 seconds walk sequence.

3.2 MAGE: Realtime and Reactive HMM-based Parameter Generation

The idea of exploring is intrinsically related to the *interactivity* of the medium. Indeed the explorative experience can only be realised if the medium – in this case the motion stylistics – is able to accept user interaction, meaning that the parameter generation process reacts accordingly. Although the MLPG algorithm is primarily based on a "all-at-once" approach, our team has been working for several years on a new short-term MLPG (ST-MLPG) algorithm (Astrinaki et al., 2012). The ST-MLPG starts to generate the trajectories in realtime, i.e. as soon as the first class label is parsed. We showed that this algorithm preserves the required naturalness and smoothness of the generated trajectories, while offering the opportunity to alter the underlying statistics on-the-fly.

The new ST-MLPG algorithm and the overall idea of reactive HMM-based synthesis led us to release a modified version of the HTS software, called MAGE (Astrinaki et al., 2011). The MAGE software library has been designed to be user-friendly and accessible for non-experts in HMMs. However the existing version was only available for speech synthesis. This work on gait modelling enabled us to get MAGE to be completely compliant with mocap data, and therefore be the first realtime SHMM-based motion synthesiser.

3.3 Creating a "Stylistic Space" with Model Interpolation/Extrapolation

The ability to compute the ST-MLPG included in MAGE on our motion models and alter the parameters of those models – i.e. means and variances – in realtime gives a first reasonable glance at the exploration of motion styles. Indeed, in our gait modelling use case, it means that we can switch instantaneously between various styles, compare them, slow them down, etc. However a very important feature is still missing in order to truly present a "stylistic space" to the user: the ability to continuously travel between (interpolate) and beyond (extrapolate) existing styles.

This notion of *model interpolation* exists in the HTS research – hence for speech modelling – for quite some time (Zen et al., 2009). The idea is to train K differentiated models λ_k. We call λ_0 the average model, i.e. the model trained on all the available data, and each k then corresponds to a new model, trained only on the data of a given style. For instance, in our gait modelling use case, λ_1 would be proud, λ_2 decided, etc. Then we can build a new model $\lambda^\star$ in which each parameter is the weighted sum of all the same parameters taken in the K models λ_k (or a subset). Both duration and observation models are modified in the computing of the parameters of $\lambda^\star$.

Depending on the strategy behind the weighted sum, the resulting $\lambda^\star$ model can be (see Figure 6):

- the *blending* between two or more arbitrary styles;

- the *inhibition* of a given style: interpolation between a style λ_k and the average model λ_0;
- the *exaggeration* of a given style: interpolation between a style λ_k and the average model λ_0, extrapolating beyond the style λ_k;
- the *inversion* of a given style: interpolation between a style λ_k and the average model λ_0, extrapolating beyond the average model λ_0.

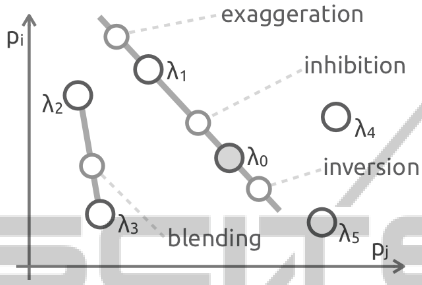

Figure 6: Illustration (on two parameters) of how a weighted sum between K differentiated models λ_k can help to create inhibition, exaggeration, inversion of a given style, and also blending between two or more arbitrary styles.

Model interpolation based on weighted sums was already available for speech synthesis in the existing version of MAGE (Astrinaki et al., 2012). Along with adapting the computation of the MLPG for mocap data, we have also adapted the model interpolation routines. As a result, we can now load our whole collection of HTS models trained on various motion styles in MAGE and interactively change the weights between the parameters of those models, so as to create the $\lambda^\star$ model that is required by the MLPG. This feature ultimately creates an interactive stylistic space based on motion statistics. Indeed the user action of inhibiting, exaggerating, inverting a style or blending between arbitrary styles will instantaneously be converted into the ongoing animation trajectories to be applied on the virtual character.

4 APPLICATION DESIGN

Section 3 has described the series of improvements that we have achieved on the MAGE software library. In this Section, we show how we have integrated this new version of MAGE in an end-user application for the exploration of motion stylistics. In the design of this application, we wanted the user to be able to carefully *mix* styles, like a musician mixes sounds from different tracks. Therefore we have developed a standalone “console-like” application that sits on the top of Blender (Blender, 2002) and provides a layer of stylistic abstraction to the user, while sending animation trajectories in realtime to the virtual character.

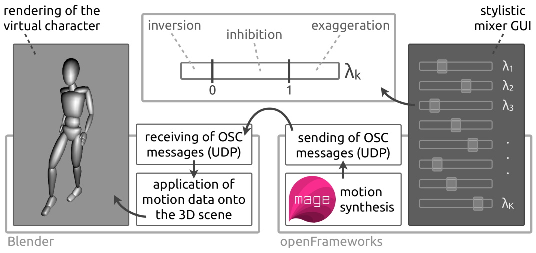
rendering of the virtual character
inversion
inhibition
exaggeration
0
1
λk
stylistic mixer GUI
λ1
λ2
λ3
.
.
.
λK
receiving of OSC messages (UDP)
sending of OSC messages (UDP)
application of motion data onto the 3D scene
mage
motion synthesis
Blender
openFrameworks

Figure 7: Summary of the design used to create our stylistic gait exploration application: Blender-rendered virtual character receiving animation data and a standalone GUI running MAGE in order to send the animation data.

In term of implementation, the application relies on an OSC communication running between Blender and a custom standalone C++ application for the model interpolation and the parameter generation algorithms as included in MAGE. A Python script has been developed in Blender so as to listen to incoming OSC messages and apply the received animation data to the virtual character in realtime. On the other side, openFrameworks has been used as our C++ application builder. MAGE has been integrated in a single-window openFrameworks project with a series of custom sliders directly connected to the weights applied on each λ_k. As described in Figure 7, each model slider could either inhibit, exaggerate or invert the model within the overall blending strategy. An illustration of reactive walk style control can be found at `http://youtu.be/OeUmVDxdJc8`.

5 DISCUSSION

The application described in Section 4 has been informally presented to a group of test users. As soon as the application is started, the virtual character walks continuously. At the starting point, the character walks in a neutral way, i.e. with no specific influence of any λ_k stylistic model on the average model. Users were proposed to play freely with the console of sliders and highlight the interesting walking styles that they would find through the exploration.

As illustrated in Figure 8, the user exploration reveals a much wider range of stylistic expression than any top-down observation of original mocap data would have suggested. By playing with features like inhibition, exaggeration and inversion of a given style, the users get a much more impersonated perspective on what has been originally recorded and statistically modelled for that style. By combining arbitrary styles, users create new walks that, if they have not

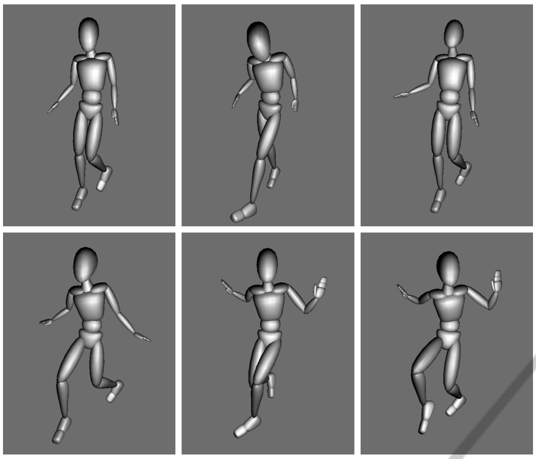

Figure 8: Examples of gait poses coming from the exploration application for various configurations of the sliders. We can see inhibited/exaggerated versions and arbitrary combinations of various styles.

been originally recorded, are however consistent with the overall stylistic behaviour of the captured subject.

With this application, users have been able to get their hands on a very high-dimension space (originally 3260 parameters) through a simplified GUI, while not reducing nor hiding its complexity and variability. Compared to playing back original mocap sequences, the ability to browse a continuous stylistic space in realtime is more interactive and user-centred. Finally the representation of motion data through a 3D virtual character helps users to experience the real motion and not a non-intuitive series of motion curves. We think this made a huge difference in users' ability to understand the ongoing stylistics.

6 CONCLUSIONS

In this paper we presented an innovative approach to the exploration of stylistic motion capture databases through a realtime motion synthesis framework. The feasibility and pertinence of this approach has been demonstrated on an expressive gait space exploration use-case. Our application enables the user to freely browse the stylistic space, exaggerate, inhibit or invert the styles present in the training data, but also to create new styles through combination of the existing styles. This reactive control provides a completely new way of visualising and exploring the motion style space. Since motion style is a notion difficult to describe or apprehend, we believe this approach to be a valuable tool for the exploration and comprehension of expert gestures which are a part of the intangible cultural heritage which is very difficult to represent.

ACKNOWLEDGEMENTS

J. Tilmanne and T. Ravet are supported by the European Union (FP7-ICT-2011-9), under grant agreement n^o 600676 (i-Treasures project). N. d'Alessandro is funded by a regional fund called Région Wallonne FIRST Spin-Off. M. Astrinaki is supported by a PhD grant funded by UMONS and Acapela Group.

REFERENCES

Animazoo (2008). IGS-190. http://www.animazoo.com.

Astrinaki, M., D'Alessandro, N., Picart, B., Drugman, T., and Dutoit, T. (2012). Reactive and Continuous Control of HMM-Based Speech Synthesis. In *IEEE Workshop on Spoken Language Technology*.

Astrinaki, M., Moinet, A., and D'Alessandro, N. (2011). MAGE: Reactive HMM-Based Software Library. http://mage.numediart.org.

Blender (2002). Blender. http://www.blender.org.

Brand, M. and Hertzmann, A. (2000). Style Machines. In *27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 183–192.

Grassia, F. S. (1998). Practical Parameterization of Rotations Using the Exponential Map. *Journal of Graphics Tools*, 3(3):29–48.

Li, Y., Wang, T., and Shum, H. Y. (2002). Motion Texture: a Two-Level Statistical Model for Character Motion Synthesis. In *29th Annual Conference on Computer Graphics and Interactive Techniques*, pages 465–472.

Tilmanne, J., Moinet, A., and Dutoit, T. (2012). Stylistic Gait Synthesis Based on Hidden Markov Models. *EURASIP Journal on Advances in Signal Processing*, 2012:72(1):1–14.

Tilmanne, J. and Ravet, T. (2010). The Mockey Database. http://tcts.fpms.ac.be/~tilmanne/.

Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., and Kitamura, T. (2000). Speech Parameter Generation Algorithms for HMM-Based Speech Synthesis. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1315–1318.

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

Wang, Y., Xie, L., Liu, Z., and Zhou, L. (2006). The SOMN-HMM Model and Its Application to Automatic Synthesis of 3D Character Animation. In *IEEE Conference on Systems, Man, and Cybernetics*, pages 4948–4952.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T. (1998). Duration Modelling for HMM-Based Speech Synthesis. In *5th International Conference on Spoken Language Processing*, pages 29–32.

Zen, H., Tokuda, K., and Black, A. W. (2009). Statistical Parametric Speech Synthesis. *Speech Communication*, 51(11):1039–1064.