J Multimodal User Interfaces
DOI 10.1007/s12193-008-0009-x

ORIGINAL PAPER

An audio-driven dancing avatar

Ferda Ofli · Yasemin Demir · Yücel Yemez · Engin Erzin · A. Murat Tekalp · Koray Balcı · İdil Kızoğlu · Lale Akarun · Cristian Canton-Ferrer · Joëlle Tilmanne · Elif Bozkurt · A. Tanju Erdem

Received: 24 December 2007 / Accepted: 5 March 2008
© OpenInterface Association 2008

Abstract We present a framework for training and synthesis of an audio-driven dancing avatar. The avatar is trained for a given musical genre using the multicamera video recordings of a dance performance. The video is analyzed to capture the time-varying posture of the dancer's body whereas the musical audio signal is processed to extract the beat information. We consider two different marker-based schemes for the motion capture problem. The first scheme uses 3D joint positions to represent the body motion whereas the second uses joint angles. Body movements of the dancer are characterized by a set of recurring semantic motion patterns, i.e., dance figures. Each dance figure is modeled in a supervised manner with a set of HMM (Hidden Markov Model) structures and the associated beat frequency. In the synthesis phase, an audio signal of unknown musical type is first classified, within a time interval, into one of the genres that have been learnt in the analysis phase, based on mel frequency cepstral coefficients (MFCC). The motion parameters of the corresponding dance figures are then synthesized via the trained HMM structures in synchrony with the audio signal based on the estimated tempo information. Finally, the generated motion parameters, either the joint angles or the 3D joint positions of the body, are animated along with the musical audio using two different animation tools that we have developed. Experimental results demonstrate the effectiveness of the proposed framework.

Keywords Multicamera motion capture · Audio-driven body motion synthesis · Dancing avatar animation

F. Ofli (✉) · Y. Demir · Y. Yemez · E. Erzin · A.M. Tekalp
Multimedia, Vision and Graphics Laboratory, Koç University,
İstanbul, Turkey
e-mail: fofli@ku.edu.tr

Y. Demir
e-mail: ydemir@ku.edu.tr

Y. Yemez
e-mail: yyemez@ku.edu.tr

E. Erzin
e-mail: eerzin@ku.edu.tr

A.M. Tekalp
e-mail: mtekalp@ku.edu.tr

K. Balcı · İ. Kızoğlu · L. Akarun
Multimedia Group, Boğaziçi University, İstanbul, Turkey

K. Balcı
e-mail: koraybalci@boun.edu.tr

İ. Kızoğlu
e-mail: idilkizoglu@boun.edu.tr

L. Akarun
e-mail: akarun@boun.edu.tr

C. Canton-Ferrer
Image and Video Processing Group, Technical University of
Catalonia, Barcelona, Spain
e-mail: ccanton@gps.tsc.upc.edu

J. Tilmanne
TCTS Lab, Faculty of Engineering of Mons, Mons, Belgium
e-mail: joelle.tilmanne@fpms.ac.be

E. Bozkurt · A.T. Erdem
Momentum Digital Media Technologies, İstanbul, Turkey

E. Bozkurt
e-mail: ebozkurt@momentum-dmt.com

A.T. Erdem
e-mail: terdem@momentum-dmt.com

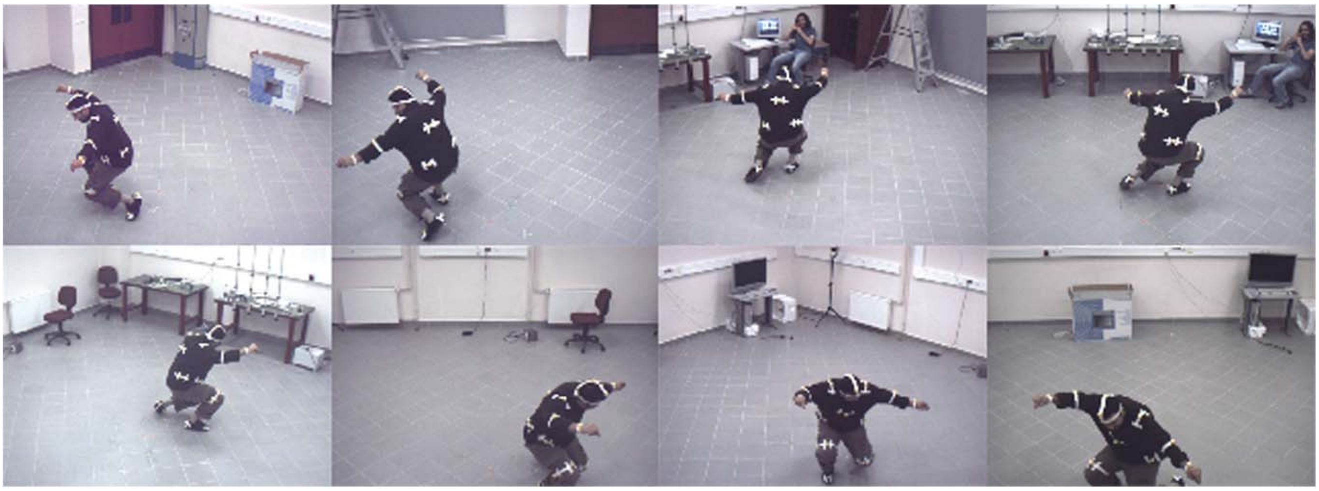


Fig. 1 An example scene captured by the multicamera system available at Koç University. Markers are attached at or around the joints of the dancer's body

1 Introduction

In a typical dance performance, the body movements of the dancer are primarily driven by, and hence, highly correlated with the musical audio signal. The work presented in this paper can be thought of as a first attempt to model this correlation towards the goal of automatic synthesis of a dancing avatar driven by musical audio.

In the signal processing literature, there exists little research that addresses the problem of audio-driven human body motion synthesis. The most relevant literature is on speech-driven lip animation [1]. Since lip movement is physiologically tightly coupled with acoustic speech, it is relatively an easy task to find a mapping between the phonemes of speech and the visemes of lip movement. Many schemes exist to find such audio-to-visual mappings among which the HMM (Hidden Markov Model)-based techniques are the most common as they yield smooth animations exploiting temporal dynamics of speech. Some of these works also incorporate synthesis of facial expressions along with the lip movements to make animated faces look more natural [2–5]. The more recent works that study the correlation between head gestures and speech prosody [6] or between hand gestures and speech content [7] towards the goal of more realistic speaker animation can also be considered in the same context.

The analysis and synthesis of body movements driven by musical audio pose more difficult challenges as compared to the speaker animation problem. In the first place, the body motion patterns, i.e., the dance figures, are usually very complicated in structure, having certain syntactic rules and hierarchies of figures. They are open to interpretation, and exhibit variations in time even for the same person. Secondly, the characteristic features of a musical audio signal, such as beat, tempo and tune, that are important in driving the dance performance are not well defined and hence need to be studied from the signal processing perspective.

We address the audio-driven body motion analysis and synthesis problem considering the most simplistic scenario possible in a dance performance. Figure 1 demonstrates our general setting for this scenario. Our dancing avatar automatically classifies the genre of a given musical piece and associates with it a single dance figure that it learns from manually segmented multiview video sequences of the dancer. Each dance figure is modeled and synthesized using an HMM structure, and synchronized with the musical audio signal using the beat information. A crucial task during avatar training is capturing the motion of the dancer. Two different marker-based tracking techniques, one of which is based on annealing particle filtering and a major contribution of this work, are employed for this purpose.

2 System overview

The overall system, as depicted in Fig. 2, comprises three modules: multimodal analysis (training), audio-driven body motion synthesis and animation. In the analysis block, multiview video sequences are analyzed in order to capture the time-varying posture of the dancer's body while audio is processed to extract beat information. Two different feature sets are considered as body posture parameters, i.e., joint angles and 3D joint positions. A marker-based motion capture system is employed to extract these feature sets. For analysis of motion features, the multiview videos are manually segmented into semantic recurring motion patterns: the dance figures. The corresponding body posture parameters are then used to train a set of HMMs, each modeling a different dance figure. Since the audio and video sequences are

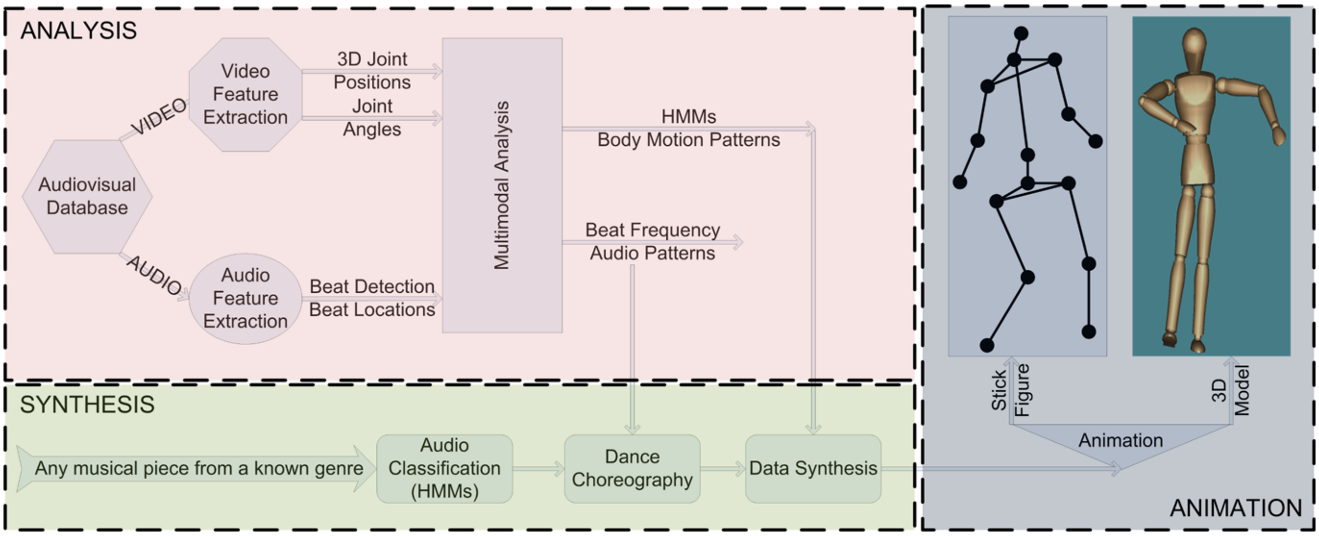


Fig. 2 Block diagram of the complete analysis-synthesis system

synchronized, each repetition of a dance figure determines a time segment from which the beat frequency associated with the figure can be estimated.

In the synthesis module, a given musical audio signal is first classified, within a time interval, into one of the genres that have been learnt in the analysis part. For genre classification, we rely on mel frequency cepstral coefficients (MFCC) and employ the HMM-based classification technique described in [8]. Beat information is then extracted in order to decide on the dance figure to synthesize and its duration. Afterwards, the system generates the body motion parameters associated with the chosen dance figures by using the corresponding HMMs, in synchrony with the beat information.

Finally, the motion parameters, either the joint angles or the 3D joint positions of the body posture, are animated using two different animation tools that we have developed. We animate the set of joint positions on a 3D stick figure and the set of joint angles on a 3D human body model.

Currently, our avatar has been trained to classify and dance only two genres, *salsa* and *belly*, and is capable of making a single dance figure associated to each genre.

3 Multicamera body motion tracking

A marker based approach is employed for motion tracking where a set of distinguishable color markers are attached at or around the joints of the dancer. There exist a number of marker-based commercial systems as evaluated in [9, 10] for human motion capture but most of them rely on a high number of cameras to avoid occlusions, high frame rates or expensive hardware. In this work, we describe two low-cost methods for multi-camera marker-based body motion capture, that is accurate enough to train our dancing avatar.

The first method tracks the 3D positions of the joints of the body based on the markers' 2D projections on each camera's image plane. The second method uses the angles at the joints of the body as posture parameters and tracks them based on annealing particle filtering using the markers' 2D projections on each camera's image plane. The former method allows users to intervene into the tracking process, and therefore, has the benefit of producing accurate tracking results by letting users correct errors manually during the tracking process. However, the tracking process itself may become lengthy and cumbersome process. The latter method, on the other hand, is automatic and eliminates the necessity of user intervention into the tracking process. This simplifies the overall motion capture process at an acceptable cost of accuracy loss.

3.1 Initialization

For a given frame in the video sequence, a set of N images are obtained from the N cameras. Each camera is modeled using a pinhole camera model based on perspective projection. Accurate calibration information is available. In order to estimate the 2D positions of the markers attached to the body of the dancer in the set of N images for a given frame, the original images are processed in the YCrCb color space which gives flexibility over intensity variations in the frames of a video as well as among the videos captured by the cameras from different views. In order to learn the chrominance information of the marker color, markers on the dancer are manually labeled in the first frame for all camera views. We assume that the distributions of Cr and Cb channel intensity values belonging to marker regions are Gaussian. Thus, we calculate the mean, μ, and the covariance, $\boldsymbol{\Sigma}$, over each marker region (a pixel neighborhood around the labeled point), where $\mu = [\mu_{Cr}, \mu_{Cb}]^T$ and $\boldsymbol{\Sigma} = (\mathbf{c} - \mu)(\mathbf{c} - \mu)^T$, $\mathbf{c}$ being $[c_{Cr}, c_{Cb}]^T$. Then, a threshold in the Mahalanobis sense with $(\mu, \boldsymbol{\Sigma})$ is applied to all images in order to detect marker locations. The number of detected markers in every image may vary due to occlusions. However, tracking information and redundancy among views allow us to overcome this problem.

3.2 Tracking the 3D joint positions

The motion capture process in this case involves retrieving the body configuration in terms of its defining parameters, namely $\mathbf{P}_t = \{p_0, \ldots, p_{M-1}\}_t$, from the multiple video streams at a given time t. This set of parameters includes the 3D positions of the markers located about the articulation points. The 3D position of each marker at each frame is determined via triangulation based on the observed 2D projections of the markers on each camera's image plane.

Let M bethe number of markers on the dancer and $\mathbf{W}$ be the set of search windows, where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M]$ such that each window $\mathbf{w}_m$ is centered around the location, $[x_m, y_m]^T$, of the corresponding marker. The set $\mathbf{W}$ is used to track markers over frames. Thus the center of each search window, $\mathbf{w}_m$, is initialized as the point manually labeled in the first frame and specifies the current position of the marker.

To track the marker positions through the incoming frames, we use the Mahalanobis distance from $\mathbf{c}$ to $(\mu, \boldsymbol{\Sigma})$ where $\mathbf{c}$ is a vector containing Cr and Cb channel intensity values $[c_{Cr}, c_{Cb}]^T$ of a point $\mathbf{x}_n \in \mathbf{w}_m$. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L]$ be the set of candidate pixels for which the chrominance distance is less than a certain threshold. If the number of these candidate pixels, L, is larger than a predefined value, then we label that marker as visible in the current camera view and update its position as the mean of the points in $\mathbf{X}$ for the current camera view. The same process is repeated for all marker points in all camera views. Hence, we have the visibility information of each marker from each camera, and for those that are visible, we have the list of 2D positions of the markers on that specific camera image plane.

Once we scan the current scene from all cameras and obtain the visibility information for all markers, we start calculating the 3D positions of the markers by back-projecting the set of 2D points which are visible in respective cameras, using triangulation method. Theoretically, it is sufficient to see

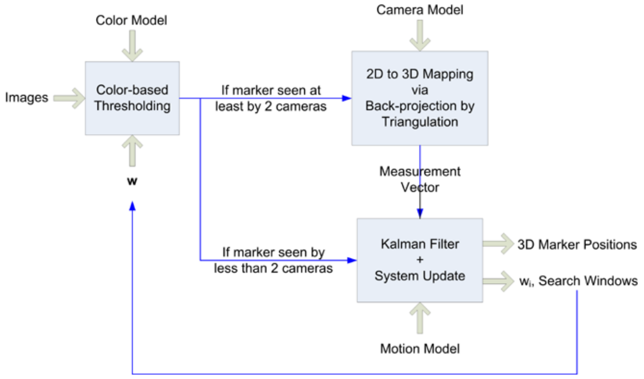

Color Model
Camera Model
Images
Color-based Thresholding
If marker seen at least by 2 cameras
2D to 3D Mapping via Back-projection by Triangulation
W
Measurement Vector
If marker seen by less than 2 cameras
Kalman Filter + System Update
3D Marker Positions
w_i, Search Windows
Motion Model

Fig. 3 Block diagram of the proposed 3D joint positions tracking system

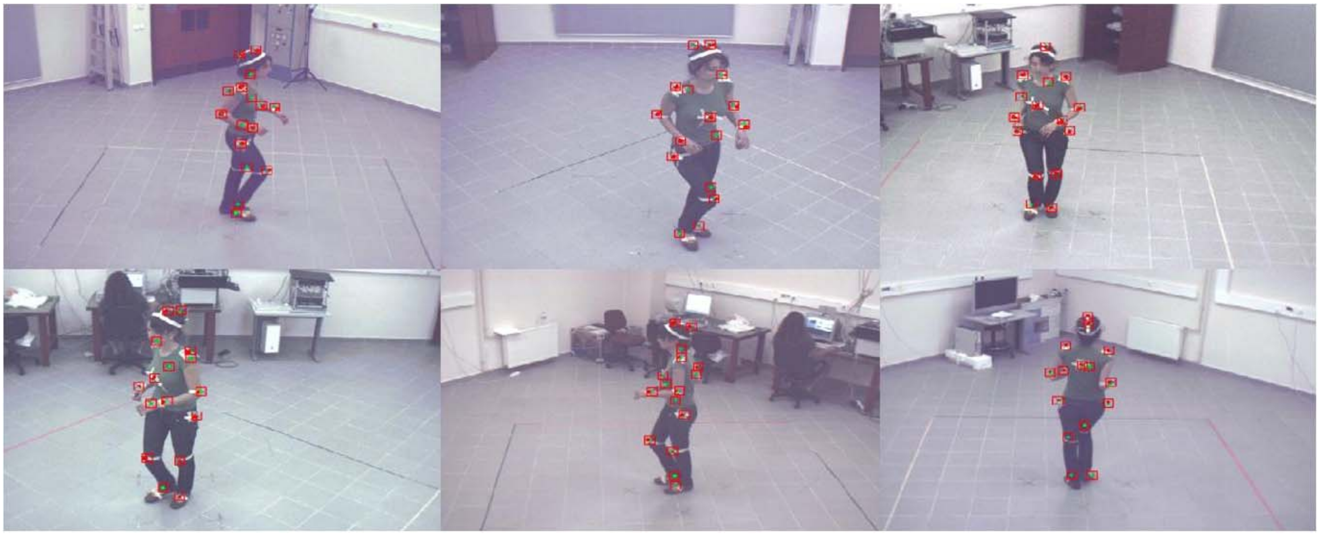


Fig. 4 An example scene from the 3D joint positions tracking process. *Red pixel* regions in the red search windows represent the marker candidate pixels for the current frame. *Green dots* are the 2D projections of the 3D marker positions for the previous frame

a marker at least from two cameras to be able to compute its position in 3D world. If a marker is not visible at least from two cameras, then its current 3D position is estimated from the information in the previous frame.

The 3D positions of markers are tracked over frames by Kalman filtering where the filter states correspond to 3D position and velocity of each marker. The list of 3D points obtained by back-projection of visible 2D points in respective camera image planes constitutes the observations for this filter. This filtering operation has two purposes:

- to smooth out the measurements for marker locations in the current frame,
- to estimate the location of each marker in the next frame and to update the positioning of each search window, $\mathbf{w}_m$, on the corresponding image plane accordingly.

Figure 3 summarizes the overall system for tracking 3D joint positions. Having updated the list of 3D joint positions for the current frame and estimated the location of the search windows for the next frame, we move on to the next frame and search the marker positions within the new search windows. This algorithm is repeated for the whole video. An instance of the 3D joint positions tracking process is shown in Fig. 4.

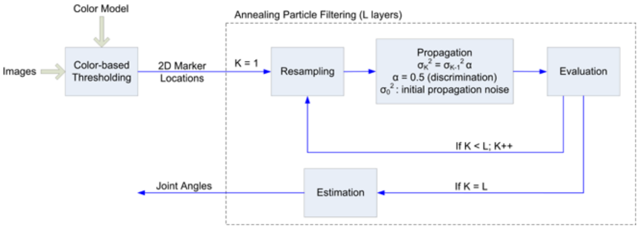

Color Model
Annealing Particle Filtering (L layers)
Images
Color-based Thresholding
2D Marker Locations
K = 1
Resampling
Propagation
$\sigma_K^2 = \sigma_{K-1}^2 \alpha$
α = 0.5 (discrimination)
σ_0^2 : initial propagation noise
Evaluation
If K < L; K++
Joint Angles
Estimation
If K = L

Fig. 5 Block diagram of the proposed system for tracking 3D joint positions

3.3 Tracking the joint angles

The set of body posture parameters, in this case, includes the articulation angles, $\mathbf{\Theta}_t = \{\theta_0, \ldots, \theta_{M-1}\}_t$, along with torso rotation and translation. The general scheme for extracting the joint angles from the set of 2D marker positions is summarized in Fig. 5.

In order to analyze the incoming data, i.e., the set of 2D marker locations for N views, an articulated body model is employed. This body model allows exploiting the underlying antropomorphic structure of the data [11]. The employed model is formed by a set of joints and links representing the limbs, head and torso of the human body and a given number of degrees of freedom (DoF) are assigned to each articulation (joint). Particularly, our model has 22 DoFs to properly capture all possible movements of the body (see an example of this in Fig. 6).

We track the body angles $\mathbf{\Theta}_t$ along time using an Annealing Particle Filtering strategy [12]. This technique is employed to tackle estimation problems involving a high dimensional state space such as in this articulated human body tracking task. Two major issues must be addressed when employing particle filtering: likelihood evaluation and propagation model. The first establishes the observation model, that is, how a given configuration of the body matches the incoming data. For a given particle, we compute the 3D positions of the articulations by means of exponential maps [11] and then project them onto the N incoming images. In order to compute the likelihood of the detected markers against the projected position of the joints, we employ the robust symmetric epipolar distance introduced in [13]. This distance measures the closeness of a set of 2D points observed as the projections of the same 3D location from different views, exploiting the redundancy among cameras.

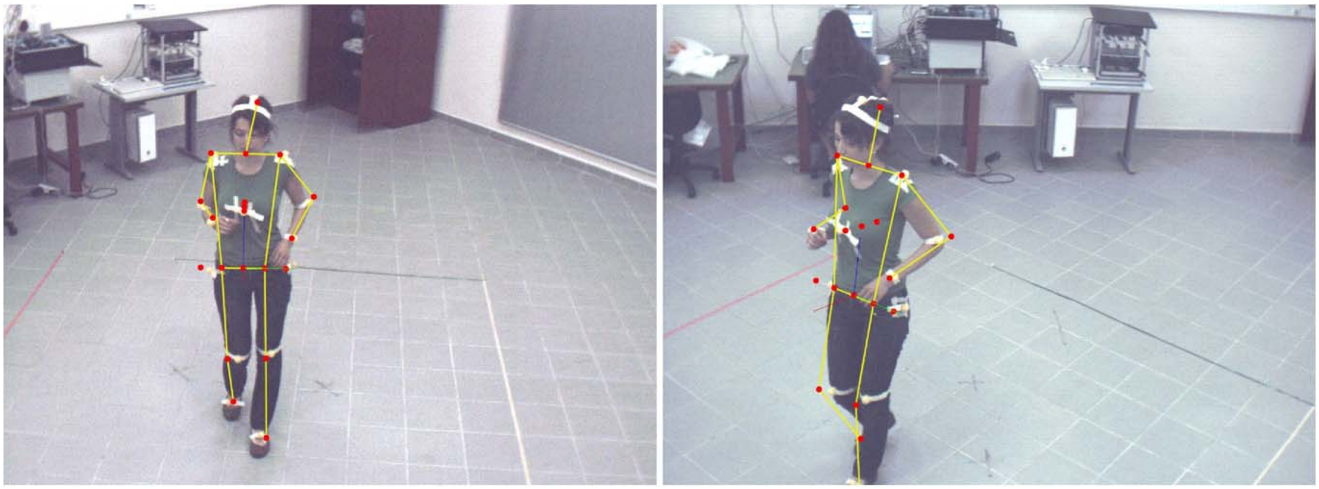


Fig. 6 An instance of the marker-based human body motion tracking process from two camera views. The articulated body model with 22 DoFs is represented as a stick model on the dancer's body

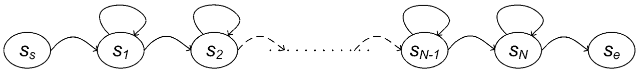


Fig. 7 A simple left-to-right HMM structure with N states in between the start and end states

The propagation model is adopted to add a drift to the angles of the particles in order to progressively sample the state space in the following iterations [14]. Moreover, an underlying motion pattern is employed in order to efficiently sample the state space, thus reducing the number of particles required. This motion pattern is represented by the kinematical constrains and physical limits of the joints of the human body. An instance of the joint angles tracking process is shown in Fig. 6.

4 Multimodal analysis

4.1 Body motion analysis

We employ HMMs to model the dance figures, i.e., the body motion patterns recurring in the dance performance. Since we performed two different techniques for body motion tracking, we have two different sets of body motion features: joint angles and joint positions. We perform separate body motion analysis tasks using these two parameter sets individually. In the first case, the HMMs are trained with the parameter set resulting from the joint angles tracking process, that includes the joint angles as well as the rotation and translation of the torso. In the second case, the HMMs are trained with the parameter set resulting from the 3D joint positions tracking process, that basically consists of the 3D coordinates of the joints at each frame.

In both cases, the same approach is adopted for analysis to interpret the recurrent body motion patterns as the common dance figures. In the case of joint angles, three separate HMMs are employed for each dance figure to better capture the dynamic behavior of the dancing body; one for the torso and two for the upper and lower parts of the body. In the other case where 3D joint positions are used, two separate HMMs are employed for each dance figure; one for the upper and one for the lower part of the body. In both cases, the HMM structure for the upper part of the body models basically the movement of the arms while the one for the lower part models the movement of the legs. There is no need for a third HMM in the latter case because, unlike the former case, the information for torso does not need to be handled explicitly.

A typical dance figure contains a well-defined sequence of movements, hence we employ a left-to-right HMM structure to model each figure (Fig. 7). Each body posture parameter is represented by a single Gaussian function and one full covariance matrix is computed for each HMM model. This rather simple scheme leads to satisfactory results without need for more complicated HMM configurations. For all HMM-related computations, we have used the "Hidden Markov Model Toolkit" (HTK) [15].

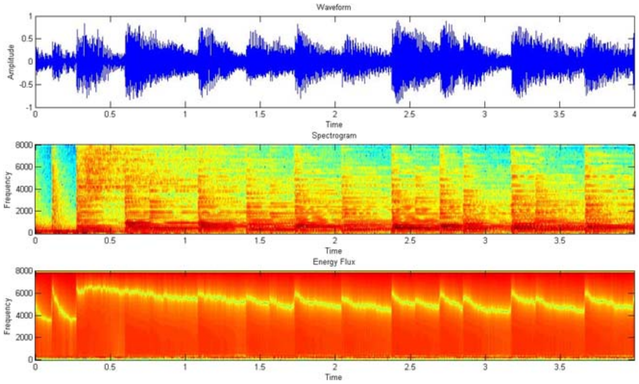


Fig. 8 Beat detection example: time waveform, spectrogram and spectral energy flux of 4 seconds of *salsa* type music computed with a 50% overlap analysis window

4.2 Audio analysis

Among various features that characterize a musical audio signal, such as tonality, harmony or melody, tempo is the one that primarily drives and synchronizes the dancing act. Hence we have employed tempo and the relevant beat information as the audio features that drive our dancing avatar. We estimate the tempo in terms of beats per minute (BPM) using the algorithm suggested in [16]. Tempo estimation involves three basic tasks: onset detection, periodicity estimation and beat location estimation. Onset detection aims to point out where musical notes start, and tempo is established by the periodicity of the detected onsets. Beat location is computed directly from periodicity estimation.

First, onsets are detected based on the spectral energy flux of the input audio signal, that signifies one of the most salient features. Onset detection is determining, since beat tends to occur at onsets. Next, the periodicity is estimated from the detected onsets using an autocorrelation based method. Once the periodicity is determined, the tempo can be calculated in terms of BPM. Finally, beat locations are estimated by generating an artificial pulse train with the estimated periodicity and by cross-correlating it with the onset sequence. Maximum values of this function marks the starting of a beat location. A sample beat extraction and localization process is given in Fig. 8.

Beat information allows estimating the tempo for each dance figure, typically ranging between 60 and 200 BPM. Analysis results of our experiments show that the average tempo is 185 BPM for *salsa* and 134 BPM for *belly*. We have also observed that a *salsa* dance figure in our training video comprises 8 beats whereas a *belly* dance figure corresponds to 3 beats. We make use of this information in the synthesis step to determine the beginning and ending frames of a dance figure.

5 Synthesis

The goal of the synthesis stage is to generate the corresponding body posture parameters synchronized with a test musical audio signal. The first task is to classify the audio signal with respect to its genre (salsa or belly in our case) over sliding windows. For this, we use MFCCs and employ the HMM-based classification technique described in [8]. The classified audio tracks are then analyzed to extract the beat and tempo information via the method explained in Sect. 4.2. The genre of the audio track determines the dance figure to be synthesized (recall that in our training video there is only one single figure associated with each genre) whereas the beat locations and the tempo information determine the duration and location of the figure. We note that the beat frequency for the same dance figure may vary within a musical audio signal or from one piece to another.

The body posture parameters corresponding to each dance figure are generated using the associated HMM structures, which are trained in the motion analysis stage (see Sect. 4.1). For each dance figure, we construct a single HMM structure by coupling the individual HMMs that are trained separately for the torso and the upper and lower parts of the body. The states of each such coupled HMM structure correspond to the motion patterns that form the dance figure. The state transition probabilities are calculated from the co-occurrence matrices of audio beat numbers and video labels. Having the state sequences and the observation probabilities that are modeled as Gaussian distributions, the body posture parameters are generated along the state sequences associated with the corresponding Gaussian distribution at each state. The dance figure boundaries are overlapped and averaged in order to generate smoother figure-to-figure transitions.

The generated body posture parameters are the sequences of either joint angles or 3D joint positions. That is, one may choose to use the set of HMMs (remember that we have three separate HMMs for the torso, and the upper and lower parts of the body) that result from joint angles analysis to synthesize a sequence of joint angles synchronized with the given test audio file. One can also do the same thing with the set of HMMs (remember that in this case we only have two separate HMMs for the upper and lower parts of the body) that result from 3D joint positions analysis instead.

Finally, the generated body posture parameters are smoothed using median filtering followed by a Gaussian low-pass filter to remove motion jerkiness within a state and in the transition from one state to another.

It is crucial to note that the use of HMMs for dance figure synthesis provides us with the ability of introducing random variations in the synthesized body motion patterns for each dance figure. These variations make the synthesis results look more natural due to the fact that humans perform slightly varying dance figures at different times for the same musical piece. Another important thing is that the use of HMMs for synthesis enables us to generate dance figures with varying durations in accordance with the beat information of the given musical audio signal.

6 Animation

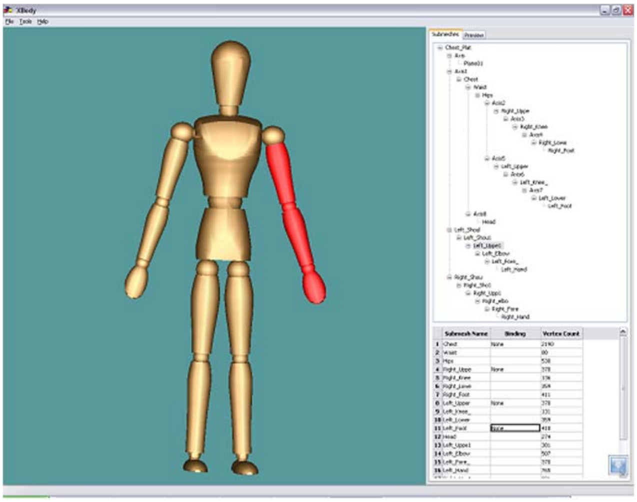
We have designed two different animation tools to visualize the output of our analysis-synthesis system. Depending on the type of the parameter set used during the analysis-synthesis process, we either animate a stick figure that is driven by a set of 3D joint positions or a 3D model that is controlled by a set of joint angles.

For stick figure animation, we developed an OpenGL based console application that is capable of animating a given set of point coordinates in 3D. The application can generate an animation of moving vertices without connecting them to each other. When the hierarchical connectivity information of the input point coordinates is available, the program generates the stick figure representation by connecting the neighboring vertices with edges. It also provides basic functionalities such as rotation, zooming in/out and panning the stick figure on the screen as well as capturing a single frame as an image or a sequence of frames as a video file. Despite depending on a simple idea, this tool proves to be useful when one wants to observe the success of the analysis-synthesis process, quickly and easily, especially in the case of 3D joint positions.

For avatar model, we use a free 3D model named Douglas F. Woodward shown in Fig. 9 with 9599 vertices and 16155 faces. The model comes with segmented hierarchy, which lets us create a kinematic chain of segments in a conventional directed acyclic graph (DAG) structure.

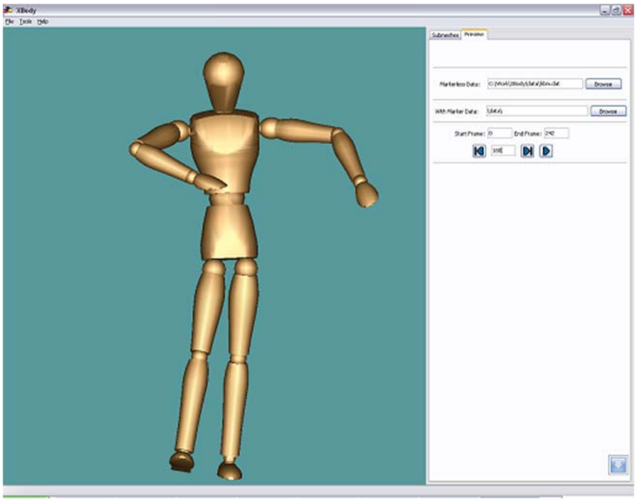
We have decided to implement a generic synthetic body representation and animation tool instead of relying on a single model. Our tool, namely Xbody, can open models in 3DS format and display the DAG and submesh info and enables labeling of the segments for animation as can be seen in Fig. 9. For rendering, Xbody relies on OpenGL and existing Xface [17] codebase. We implemented an additional forward kinematics pipeline for rendering and animation of DAG.

As for animation, the generated set of joint angles by the analysis-synthesis process can be fed to Xbody and animated with the same frame per second of video. The previewing interface of the tool enables us to inspect each frame



	Submesh Name	Binding	Vertex Count
1	Chest	None	2190
2	Waist		80
3	Hips		538
4	Right_Uppe	None	378
5	Right_Knee		176
6	Right_Lowe		359
7	Right_Foot		411
8	Left_Upper	None	378
9	Left_Knee_		331
10	Left_Lower		359
11	Left_Foot	None	418
12	Head		274
13	Left_Upper1		301
14	Left_Elbow		507
15	Left_Fore_		378
16	Left_Hand		768

Fig. 9 Xbody DAG view and labelling pane



Markerless Data: [illegible]

With Marker Data: [illegible]

Start Frame: 0 End Frame: 242

108

Fig. 10 Xbody preview pane

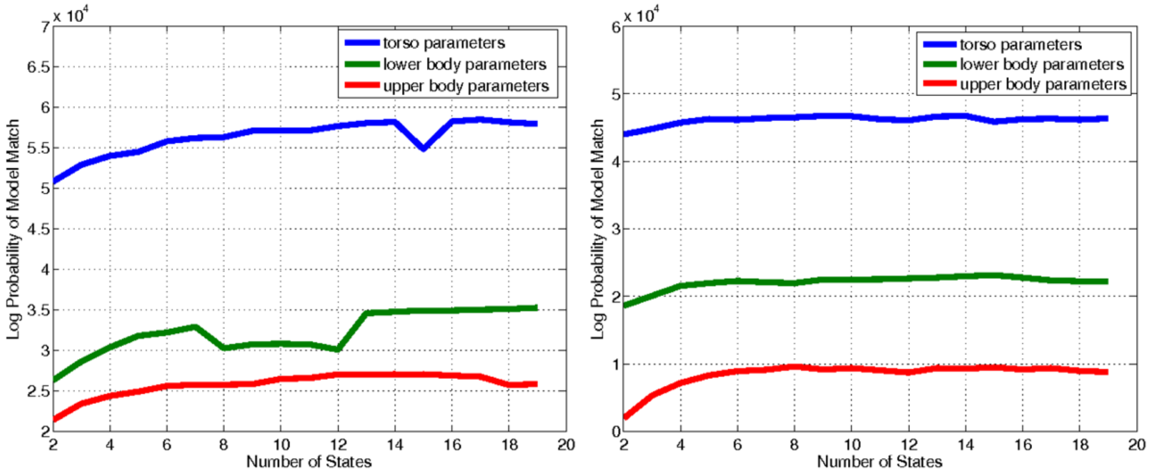


Fig. 11 Evolution of the logarithmic probability of the model match with varying number of states for the 6 HMM structures in the case of joint angles (three for *salsa* on the *left* and three for *belly* on the *right*)

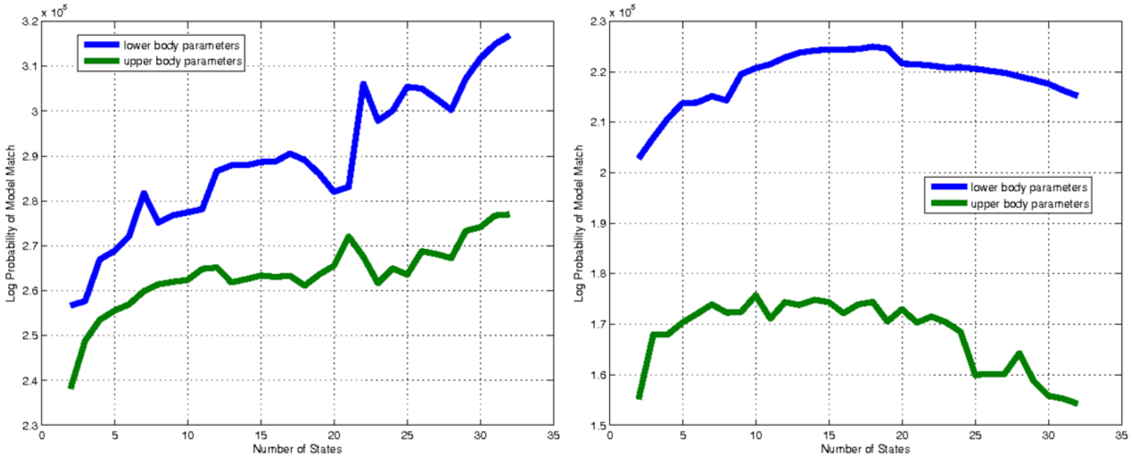


Fig. 12 Evolution of the logarithmic probability of the model match with varying number of states for the 4 HMM structures in the case of 3D joint positions (two for *salsa* on the *left* and two for *belly* on the *right*)

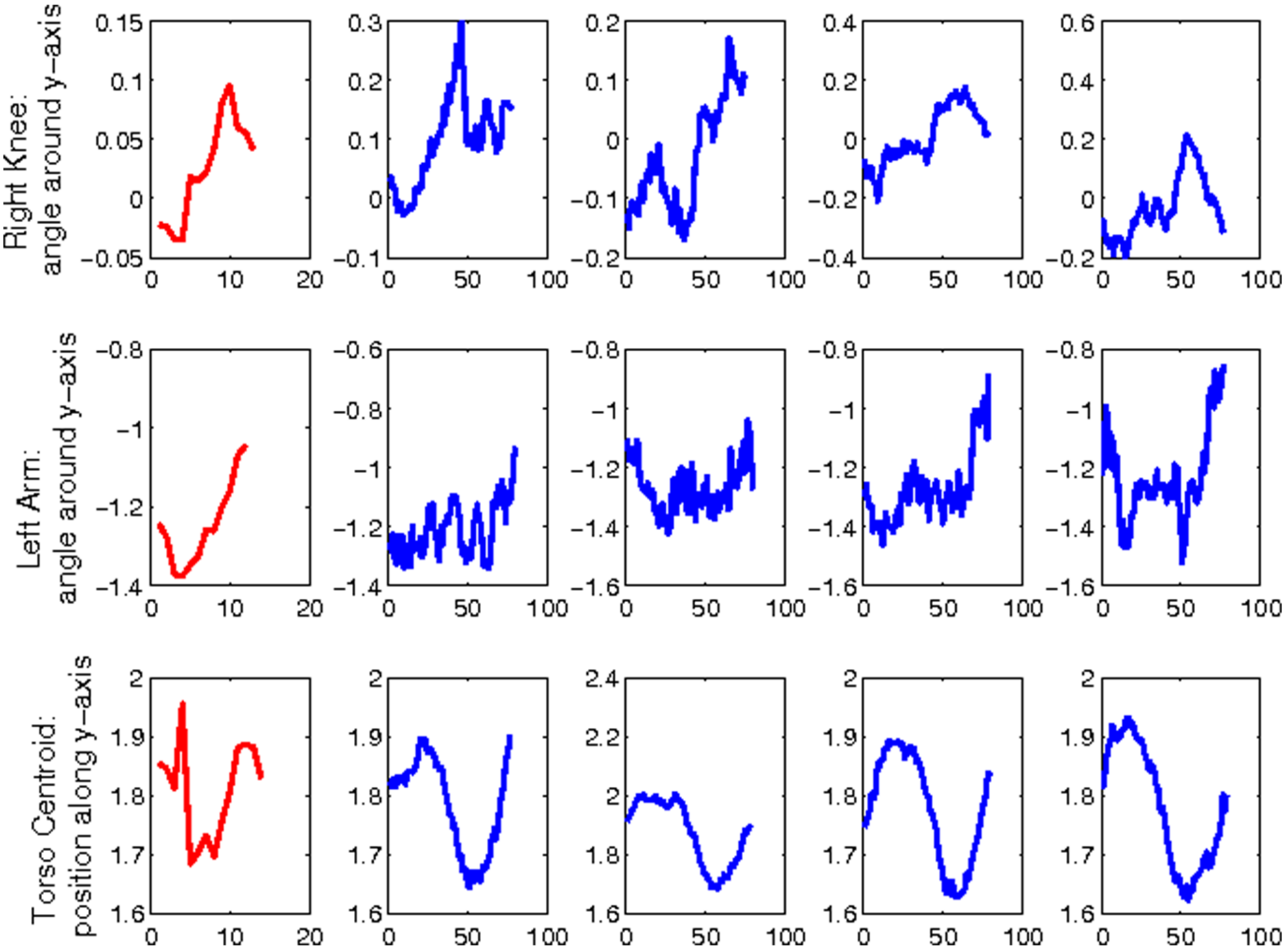


Fig. 13 For the salsa figure, variation of the means of three parameters over the HMM states (plotted in *red*) and evolution of the same three parameters during four different realizations sampled from the training video (plotted in *blue*)

by entering the frame number and using rotation, zooming in/out and panning the model on the screen. In Fig. 10, a preview of the synthesis results for *belly* dance in the case of joint angles is shown. The tool can also export the animation as a video file in AVI format.

As its current state, Xbody can be used for better analyzing the results of motion tracking algorithms and HMM based motion generation.

7 Experiments and results

Our training dataset includes multiview video recordings of two dance performances, one for *salsa* and one for *belly*, each with a duration of approximately 5 minutes. The performances are recorded synchronously from 6 cameras at 30 fps. Each video recording consists of one single dance figure repeated successively during the whole performance.

For motion analysis, we manually label the start and end frames of each dance figure throughout the entire dance recordings. Recall that we have 3 HMMs for the case of joint angles and 2 HMMs for the case of joint positions. These HMM models of each dance figure are trained in a supervised manner with the body posture parameters captured from the manually labeled segments, respectively in the case of joint angles and joint positions.

In order to determine the optimal number of states for each of the HMMs, we train each HMM with different number of states (varying from 2 to 19). By computing the average logarithmic probability of the model match for each value, we examine the progression of the learning process and the accuracy of the trained model. The evolution of this parameter in the case of joint angles for the totality of the 6 HMM structures that we trained is displayed in Fig. 11. The evolution of the same parameter in the case of joint positions for the totality of the 4 HMM structures that we trained is displayed in Fig. 12. We observe that the optimal number of states is related to the complexity of the dance figure. In the case of the salsa figure, which is more complicated than the belly, the optimal numbers are greater than

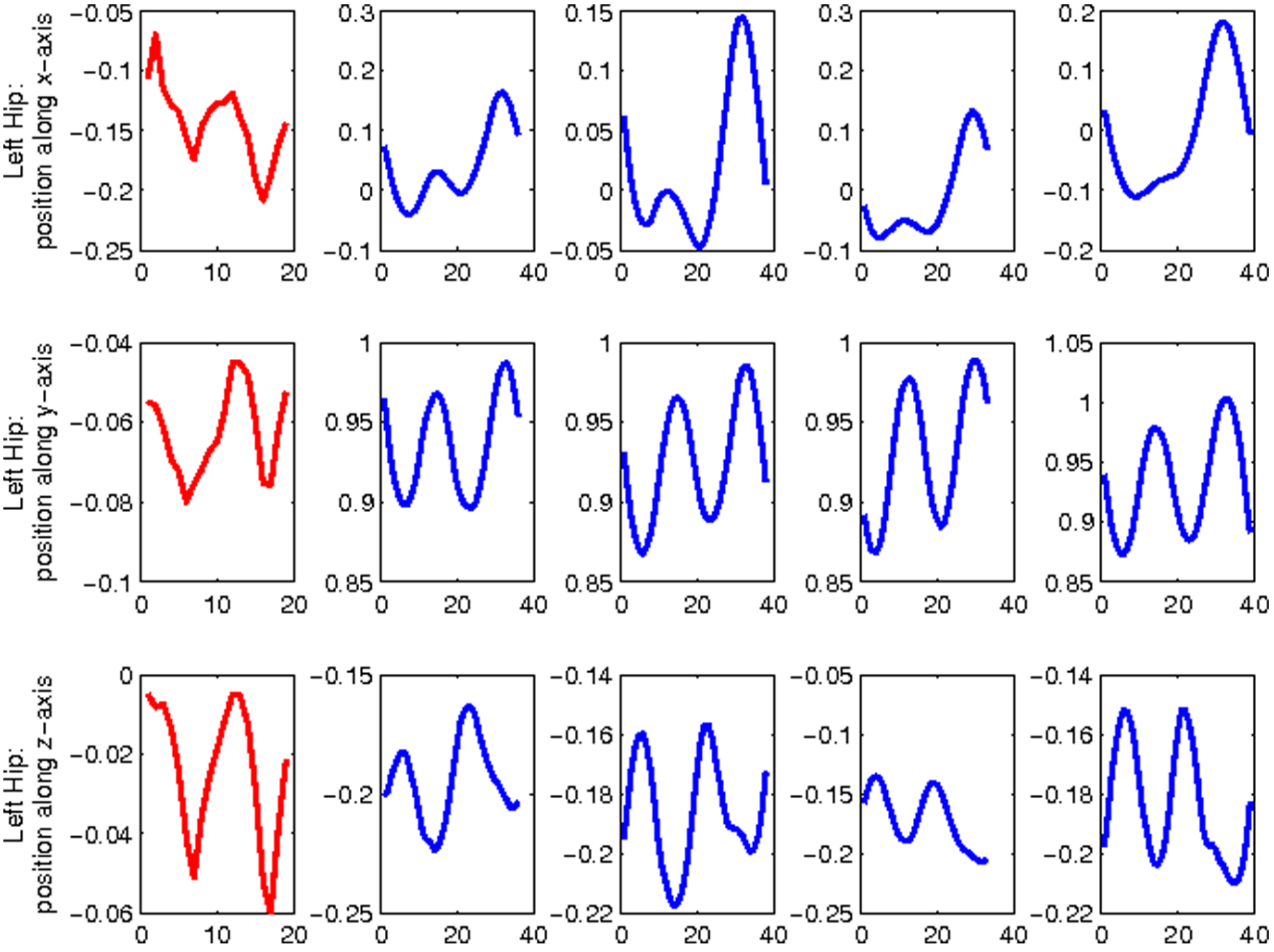


Fig. 14 For the belly figure, variation of the means of three parameters over the HMM states (plotted in *red*) and evolution of the same three parameters during four different realizations sampled from the training video (plotted in *blue*)

those for the belly figure. To determine the optimal number of states, we basically search for the peak in the plot, or the point where the plots start to saturate since we also want to keep the number of states, and hence the model complexity, as low as possible.

In order to verify that the posture parameters are correctly modeled with the resulting HMMs, in Fig. 13 and Fig. 14, we compare, for some of the parameters, the evolution of the means of their Gaussian distributions over the HMM states with the evolution of the same parameters through the realizations of the corresponding dance figures in the training data set. The shapes of the evolution are clearly observed to be similar, even for the parameters which show significant variations from one realization to another in the training set and are thus difficult to model.

The musical audio signals are recorded at 16 kHz as 16 bit mono PCM wavefiles. The signals are analyzed over a 25 ms Hamming window at every 10 ms. The set of 13 MFC coefficients along with their first and second derivatives, adding up to a total of 39 features, forms the audio feature vector for the genre classification task. Using MFCCs as the only audio feature set becomes sufficient for the classification problem in our case, since we have only two types of musical audio, *salsa* and *belly*.

We have considered several animation scenarios for demonstration of our dancing avatar. In the first scenario, we mix two audio tracks of different genres, *salsa* and *belly*, and use this mixed track as the animation audio to show that the avatar can successfully recognize the changing audio and synthesize the correct dance figures. In the second scenario, we first slow down and then speed up the audio track to demonstrate that the avatar can keep track of the changing beat information and adjust the speed of the dance movements accordingly. In the final scenario, we take an arbitrary audio which is neither salsa nor belly to see how the avatar adapts itself to a different genre that it has not been trained for. We applied these three scenarios on analysis-synthesis results of both joint angles parameter set and 3D joint positions parameter set.[1]

8 Conclusions and future work

We have developed a framework for audio-driven human body motion analysis and synthesis. We have addressed the problem in the context of dance performance and considered the most simplistic scenario possible in which only a single dance figure is associated with each musical genre. Currently, our dancing avatar has been trained for *salsa* and *belly*. The experiments show that the avatar can successfully recognize the genre changes in a given audio track and synthesize the correct dance figures in a very realistic manner. The avatar can also keep track of the changing beat information and adjust the speed of the dance movements accordingly.

A crucial task during avatar training is to capture the motion of the dancer in an accurate manner. For this, we have developed a marker-based algorithm based on annealing particle filtering, that can automatically extract the human posture from multiview video without any human intervention. We also performed an alternative marker-based tracking algorithm with human intervention. This alternative method provided us with a reference for the tracking results obtained by the algorithm based on annealing particle filtering. Nevertheless, we used both parameter sets to analyze the correlation between body posture and audio, and to synthesize body posture parameters when driven by an audio signal. We compared the analysis-synthesis results of both parameter sets on several animation scenarios by using a dancing avatar.

Our future work will involve unsupervised training of our dancing avatar for different musical genres in more complicated scenarios in which the dance figures are more sophisticated in structure, having certain syntactic rules and hierarchies of figures. To achieve this, we will also need to consider various musical audio features other than beat and tempo, such as tonality, harmony and melody.

Acknowledgements A part of this work is developed during the ENTERFACE'07 Summer Workshop on Multi-Modal Interfaces[2] and it has been supported by the European Sixth Framework Programme Network of Excellence SIMILAR,[3] by the Scientific and Technological Research Council of Turkey (TUBITAK)[4] under project EEEAG-106E201 and COST Action: 2102.[5] A. Murat Tekalp acknowledges support from Turkish Academy of Sciences (TUBA).[6]

[1] Demo videos for audio-driven dance figure analysis-synthesis system. Available at http://mvgl.ku.edu.tr/bodymotionanalysis/jmui/7.

[2] http://www.enterface.net/.

[3] http://www.similar.cc/.

[4] http://www.tubitak.gov.tr/.

[5] http://www.cost2102.eu/.

[6] http://www.tuba.gov.tr/.

References

1. Chen T (2001) Audiovisual speech processing. IEEE Signal Process Mag 18(1):9–21
2. Bregler C, Covell M, Slaney M (1997) Video rewrite: driving visual speech with audio. In: SIGGRAPH '97: Proceedings of the 24th annual conference on computer graphics and interactive techniques, New York, NY, USA. ACM Press/Addison-Wesley, New York, pp 353–360
3. Brand M (1999) Voice puppetry. In: SIGGRAPH '99: Proceedings of the 26th annual conference on computer graphics and interactive techniques, New York, NY, USA. ACM Press/Addison-Wesley, New York, pp 21–28
4. Li Y, Shum H (2006) Learning dynamic audio-visual mapping with input-output hidden Markov models. IEEE Trans Multimedia 8(3):542–549
5. Ofli F, Erzin E, Yemez Y, Tekalp AM (2007) Estimation and analysis of facial animation parameter patterns. In: IEEE International conference on image processing
6. Sargin ME, Erzin E, Yemez Y, Tekalp AM, Erdem AT, Erdem C, Ozkan M (2007) Prosody-driven head-gesture animation. IEEE Int Conf Acoustics Speech Signal Process 2:677–680
7. Sargin ME, Aran O, Karpov A, Ofli F, Yasinnik Y, Wilson S, Erzin E, Yemez Y, Tekalp AM (2006) Combined gesture—speech analysis and speech driven gesture synthesis. In: IEEE international conference on multimedia and expo, pp 893–896
8. Bagci U, Erzin E (2007) Automatic classification of musical genres using inter-genre similarity. IEEE Signal Process Lett 14:521–524
9. Ehara Y, Fujimoto H, Miyazaki S, Tanaka S, Yamamoto S (1995) Comparison of the performance of 3d camera systems. Gait Posture 3:166–169
10. Ehara Y, Fujimoto H, Miyazaki S, Mochimaru M, Tanaka S, Yamamoto S (1997) Comparison of the performance of 3d camera systems II. Gait Posture 5:251–255
11. Bregler C, Malik J (1998) Tracking people with twists and exponential maps. In: IEEE international conference on computer vision and pattern recognition
12. Deutscher J, Reid I (2005) Articulated body motion capture by stochastic search. Int J Comput Vis 61:185–205
13. Canton-Ferrer C, Casas JR, Pardàs M (2005) Towards a Bayesian approach to robust finding correspondences in multiple view geometry environments. In: Lecture notes on computer science, vol 3515. Springer, Berlin, pp 281–289
14. Arulampalam M, Maskell S, Gordon N, Clapp T (2002) A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process 50(2):174–188
15. Young S (1993) The htk hidden Markov model toolkit: design and philosophy. Technical Report TR. 153, Speech Group, Department of Engineering, Cambridge University (UK)
16. Alonso M, David B, Richard G (2004) Tempo and beat estimation of music signals. In: International conference on music information retrieval
17. Balci K, Not E, Zancanaro M, Pianesi F (2007) Xface open source project and smil-agent scripting language for creating and animating embodied conversational agents. In: MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia, New York, NY, USA. ACM Press, New York, pp 1013–1016