EAI Endorsed Transactions
on Creative Technologies
Research Article EAI.EU

Beating-Time Gestures Imitation Learning for Humanoid Robots

Denis Amelynck[1], Pieter-Jan Maes[1,*], Jean-Pierre Martens[2], Marc Leman[1]

[1]IPEM, Department of Art, Music and Theatre Sciences, Ghent University, Ghent, Belgium
[2]DSSP-ELIS, Department of Electronics and Information Systems, Ghent University, Ghent, Belgium

Abstract

Beating-time gestures are movement patterns of the hand swaying along with music, thereby indicating accented musical pulses. The spatiotemporal configuration of these patterns makes it difficult to analyse and model them. In this paper we present an innovative modelling approach that is based upon imitation learning or Programming by Demonstration (PbD). Our approach - based on Dirichlet Process Mixture Models, Hidden Markov Models, Dynamic Time Warping, and non-uniform cubic spline regression - is particularly innovative as it handles spatial and temporal variability by the generation of a generalised trajectory from a set of periodically repeated movements. Although not within the scope of our study, our procedures may be implemented for the sake of controlling movement behaviour of robots and avatar animations in response to music.

Received on 16 November 2016; accepted on 02 October 2017; published on 08 November 2017

Keywords: programming by demonstration, cubic spline regression, dynamical time warping, beating-time gestures

doi:10.4108/eai.8-11-2017.153335

1. Introduction

Robots and animated avatars have an enormous potential for technology-assisted (e-)learning purposes. Of particular interest is the ability of robots and avatars to motivate people to be active, and to assist them in learning particular motor skills. This however requires robots and avatars to accurately reproduce human behaviour. Programming by Demonstration (PbD), also referred to as Learning by Imitation, has been proven an effective method in robotics research to let robots reproduce human-like movements in a realistic manner [1, 2]. In essence, PbD enables learning a robot how to perform specific movement trajectories and task manipulations through the imitation of actions performed by a human demonstrator. However, the generation of a generalised trajectory, which can be used as a control signal, imposes some critical challenges. One particular challenge is to cope with the spatial and temporal variability inherent to repeated demonstrated actions. Temporal variability is typically solved by some form of Dynamic Time Warping (DTW) [3]. The spatial variability is then handled by interpolation techniques, such as spline fitting, resulting in smooth curves conceived as being close to natural movement.[4–7]

The main goal of the current study is to present a new, integrated PbD method to handle spatial and temporal variability in the generation of a generalised trajectory from a set of periodically repeated, multi-segmented movements. Movements in response to music are particularly relevant here as they often reflect the periodic auditory patterning of music. These movements may range from simple foot tapping or head nodding, to more complex forms of dance [8, 9]. In the current study, we focus on so-called 'beating-time gestures', which reflect the periodically repeated basic temporal pattern of strong and weak accented beats within music (i.e., 'musical meter') into a corresponding spatiotemporal 'conducting model'. Beating-time gestures have a goal-directed character, in the sense that they follow some intended spatial trajectory, and temporal key points in the music - i.e., strong and weak accented beats - are specifically linked to position, velocity, and acceleration features. Because

★Please ensure that you use the most up to date class file, available from EAI at http://doc.eai.eu/publications/transactions/latex/

*Corresponding author. Email: pieterjan.maes@UGent.be

of that goal-directed character, a conducting model is typically separated into different motion segments. The obvious method for describing beating-time gestures, which articulate a conducting model, is by looking at the shape of their trajectories. For instance for a 4/4 meter (time signature), the classical trajectory shows a movement where the right hand goes down to reach the first beat, left to reach the second beat, right to the third beat and up for the forth beat, shown as model 1 in Fig. 1. Obviously, other patterns exist and in this study we present different four-beat patterns as the ones shown in Fig. 1.

In a first part of the paper, we focus on formulating our procedure to generate a generalised beating-time gesture. This procedure will be demonstrated by an application that allows adapting a two-dimensionally represented beating-time gesture to music that is playing. In future applications, this procedure could be extended to more interesting applications, such as actual robots or avatars. Thereby, we employ a trajectory-level approach to the task, focusing on low-level features such as position and velocity. In dealing with spatial and temporal variability within repeated demonstrations of periodic, multi-segmented movements, our procedure uses a Dirichlet Process Mixture Model (DPMM) as front end for a continuous HMM (cHMM) to characterise every beating-time gesture by a set of non-equidistant key points. We use these key points for the creation of a reference signal for Dynamic Time Warping (DTW). DTW is our solution for handling the temporal variation. Eventually, we produce a smooth generalised trajectory by means of non-uniform B-cubic spline regression. The regression step accounts for the spatial variation in the set of demonstrations. As our method is well suited to handle temporal and spatial variability, it is equally well suited to capture demonstrations of models that are more difficult to perform, and hence, demonstrations that are more prone to errors or inconsistencies in time and space.

In a second part of the paper, we outline a procedure that facilitates the adaptation of a generalised beating-time gesture to music. In past studies, music-driven synthesis systems have been developed for human dance movements and choreographies [10–13]. In the context of music conducting systems, more attention has been devoted to the opposite process, namely the implementation of automatic gesture analysis to control in real-time specific playback parameters of music such as tempo, timbre, and dynamics [14–17]. In the current study, we outline a procedure that adapts beating-time gestures to music by using piece-wise linear interpolation to map the intervals between the key points of these gestures to the intervals made up by the beat points of the music . The development of such a system is highly valuable in a music-pedagogical context. Beating-time gestures are typically used in music pedagogy to reinforce musical novices' ability to perceive and identify metrical patterns in music. In support of this pedagogical method, research demonstrated that the performance of body movements in response to music may structure music and influence people's perception of musical rhythm and meter [18, 19]. Hence, a system that automatically generates beating-time gestures to music may assist students to perform these gestures themselves, and consequently to increase their musical skills.

The general outline of the paper is as follows. Section 2 describes a small experiment that was conducted to obtain human movement data on which we could apply our methods. The explanation of our data processing and modelling methods makes up the core of our study and is handled in Section 3. In Section 4, we introduce an application that enables the automatic alignment of beating-time gestures to music. This is followed by a discussion in Section 5 and conclusions are drawn in Section 6.

2. Data collection: experimental set-up

Subjects + Task. Four participants having no musical background and aged between 18 and 20 years were asked to perform repetitive cycles of conducting models into beating-time gestures. As depicted in Fig. 1 five different conducting models were defined upfront. The definition and selection of models was done in collaboration with the participants. They were asked to explore and propose a set of five models. Every subject performed 40 cycles of a particular conducting model into beating-time gestures, but did not perform on all five models. The assignment of conducting models to subjects was random with the restriction that every subject had to perform the commonly known conducting model (labeled as model 1) and two more conducting models. More in particular, subject 1 performed on model 1, 2 and 3, subject 2 performed on model 1, 2 and 4, and subjects 3 and 4 performed on model 1, 3 and 5. This means that in total 480 cycles of beating-time gestures (120 per subject) were performed.

Stimuli. The participants performed repetitive cycles of beating-time gestures on an auditory stimulus consisting of 40 bars of a repetitive metrical pattern exhibited by metronome ticks at a tempo of 120 BPM using a 4/4 time signature.

Data. Three-dimensional position data of hand movements was recorded at a sample rate of 100 Hz using an 'Optitrack' infrared optical motion capture system consisting of 12 synchronised cameras with related 'Arena' motion capture software (http://www.naturalpoint.com). Participants were asked to put on two sets of three infrared reflecting markers, each set

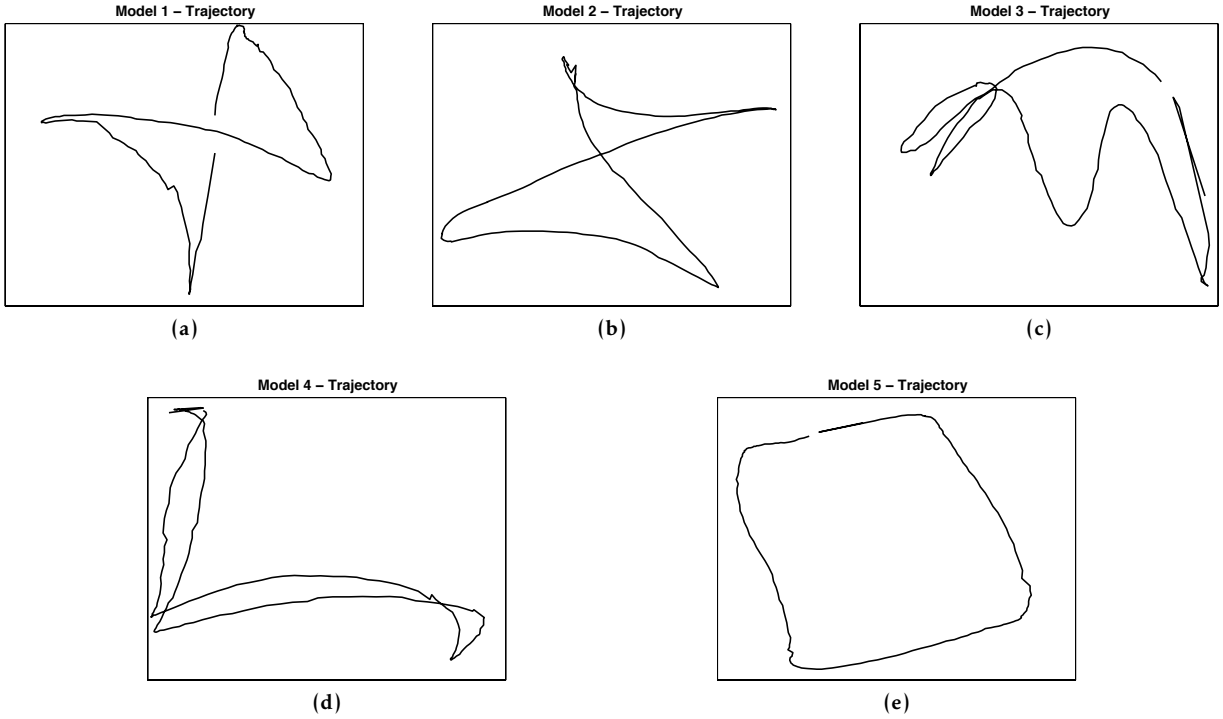


Figure 1. Spatial configuration of the five conductor models with 4/4 time signature.

defining a rigid body that could be easily identified by the motion capture software. One set was placed at the hand and one set at the chest. The set at the chest was meant for positional reference.

3. Data processing

3.1. Overview

The main goal of the study was to create a generalised trajectory that allowed - in future applications - a robot or animated avatar to perform subsequent beating-time gestures in a continuous manner in response to music. For robots, there exist many implementations (hardware and software) but a common principle is that force is applied to accomplish a positional and/or velocity target [6]. Force is normally expressed as an acceleration command (see Eq. 1) and it is used to track the desired velocity and position using a proportional-derivative (PD) controller.

$$\ddot{x} = \kappa_v(\hat{\dot{x}} - \dot{x}) + \kappa_p(\hat{x} - x) \quad (1)$$

κ_v and κ_p are gain parameters similar to damping and stiffness factors. x is a vector representing positional information in line with the degrees of freedom (DOF) of a robot. x can hold Cartesian coordinates as well as angle coordinates. We follow here the conventional notation for derivatives being $\dot{x}$ for speed and $\ddot{x}$ for acceleration. The hat-symbol is used for indicating the target values: $\hat{x}$ stands for the target position and $\hat{\dot{x}}$ for the target velocity. Eq. 1 explains the main interest of this paper. We assume that a robot can determine its current position (x) and its current velocity ($\dot{x}$). We do not discuss the details of tuning a robot (κ_v, κ_p). The focus of this paper lays completely on the calculation of $\hat{x}$ and $\hat{\dot{x}}$ or, in other words on calculating a generalised target trajectory for a beating-time gesture.

Our PbD solution calculates a generalised trajectory from a continuous series of periodic, multi-segmented beating-time gestures. The spatial variation in the series will be handled using cubic spline regression. Cubic spline regression is also an asset for handling periodic boundaries. As a beating-time gesture is part of a repeated sequence, we want the beginning and the end of the generalised gesture to coincide. Cubic spline regression is often done with a set of equidistant knots (uniform splines). Then, extrema in the trajectory can or can not coincide with the knots. If they do not coincide, the extrema of the trajectory are flattened out resulting in a more compressed shape. Because beating-time gestures use the extrema to convey beat information, we do not go that path and we choose for non-uniform splines instead. Moreover, temporal variation will be handled by adding a dynamical time warping (DTW) step. DTW warps all demonstrations non-linearly in the time dimension to a reference signal. And here, the challenge relates to the calculation of a reference signal.

Further, the issues of non-equidistant knots for cubic spline regression and the creation of a reference signal for DTW are solved by fitting a HMM. As we prefer to keep the set of demonstrations low we need a simple HMM, in our case an HMM with few parameters. The number of HMM states and the initial values for Baum-Welch training of the HMM parameters follow from a Dirichlet Process Gaussian Mixture Model (DPGMM) that we fit to the data. DPGMM is a Bayesian method using a Dirichlet process as prior. The prior acts as a regulariser, preventing overfitting and yielding models that usually generalise well. This is an asset, as in our case we have few data and model fitting with few data is prone to overfitting. For more information on DPGMM we refer to Teh [20] and El-Arini [21]. Fig. 2 gives an overview of the complete PbD procedure we propose.

3.2. Data preprocessing

Initial inspection of the data showed that in a series of performed beating-time gestures usually the first and last ones were outliers comparable to a warming-up and cooling-down effect. For that reason, these cycles were excluded from further analysis.

Beating-time gestures are simple geometric movements and most of them can be studied by projection of the positional coordinates onto the frontal (coronal) plane. In our experimental set-up the coronal plane was defined by the recorded chest markers. The coordinates of the hand markers, making up the beating-time gesture, were then orthogonally projected onto this coronal plane. This three to two dimensional reduction permits a better visualisation. For the purpose of controlling an actual robot, full dimensional data should be used instead.

In addition to the positional coordinates, we calculated the velocity as the derivative of the positional data. A local (linear) regression filter was applied to calculate smooth derivatives. The size of the regression window was set to 0.100 s corresponding with an amplitude response that is proportional to the frequency in the useful frequency band of 0-6 Hz. The 0-6 Hz range was derived from spectrograms. This regression filter was applied to all coordinates.

In the course of a demonstration, we spotted that position and size of the basic gestures changed from measure to measure. To overcome this issue we added a normalisation step. Two different methods for normalisation were considered giving slightly different results (Fig. 3). One method interprets the entire set of demonstrations as one long lasting gesture. Normalisation is then equal to high pass filtering (detrending) followed by scaling. Another method views the entire set as a sequence of separate individual basic gestures and normalises per basic gesture by subtracting the basic gesture's average value. Visual inspection learns that the latter fitted better to reality. So we opted for the second method.

The normalised variables were then stored in a four-dimensional data vector: $X_{m,n} = [posx_{m,n}\ posy_{m,n}\ velx_{m,n}\ vely_{m,n}]$ where m is the basic gesture index (in our case a value from 1 to 40) and n the sample index in our basic gesture (in our case n ranged from 1 to 200, the number of samples per gesture). We then used the notation X_m to refer to all samples from one basic beating-time gesture. In Fig. 4 we display the variables $posx, posy, velx, vely$ representing the normalised versions of the horizontal position, respectively the vertical position, the horizontal velocity and the vertical velocity.

The next steps in our solution follow roughly the procedure explored by Vakanski et al. [22] and Aleotti et al. [5] but with some adaptations to our specific needs. As our gestures are beating-time gestures, they are subject to temporal constraints imposed by the temporal structure of the music they were performed on. Additionally, our solution uses a DPGMM in combination with a continuous HMM (cHMM) opposed to the combination of Linde-Buzo-Gray-clustering and discrete HMMs adopted in [22]. The three next steps are: (i) point extraction using HMMs (section 3.3) and (ii) time warping using DTW (section 3.4) and (iii) generalised trajectory generation via non-uniform B-Spline regression (section 3.5).

3.3. Key point extraction

The key points constitute the fingerprint of a gesture, the minimum amount of information needed to reconstruct a trajectory. Our approach places the key points at the hidden state transitions of a continuous HMM (cHMM).

We consider our movement trajectories in an augmented feature space of four dimensions (4D), having two-dimensional (2D) position variables, and 2D velocity variables. Remember from Eq. 1 that we need a target trajectory for position and velocity. The number of internal HMM states and the initial values of the HMM parameters are calculated from a DPGMM. The DPGMM is similar to a Gaussian Mixture Model (GMM) except that the number of clusters is determined directly from the data and not from an additional data validation step. The DPGMM clusters are shown in Fig. 5 using two separate 2D representations, one for the positional coordinates and one for the velocity values. The cluster assignment reveals that the performance can be understood as a chain of single Gaussians. We therefore propose as model a Bakis left-to-right HMM with single Gaussian emissions.

The exact number of states is derived from a single basic gesture m^* which is selected via some criterion.

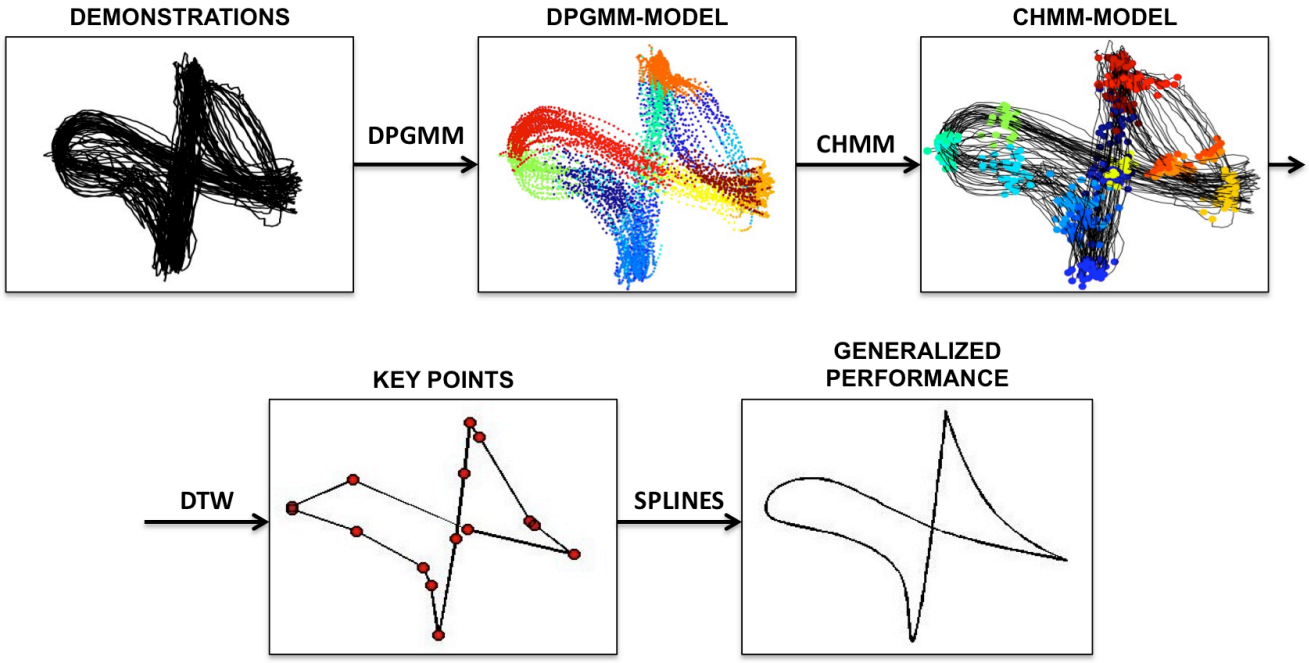


Figure 2. Flow Chart of the followed procedure for PbD. (i) DPGMM, as front-end for a continuous HMM, is used to calculate its number of hidden states and to set its initial emission values. (ii) a cHMM defines the key points which are used to create a reference signal for (iii) DTW. Eventually (iv) non-uniform cubic spline regression on the warped gestures produces a smooth generalised gesture.

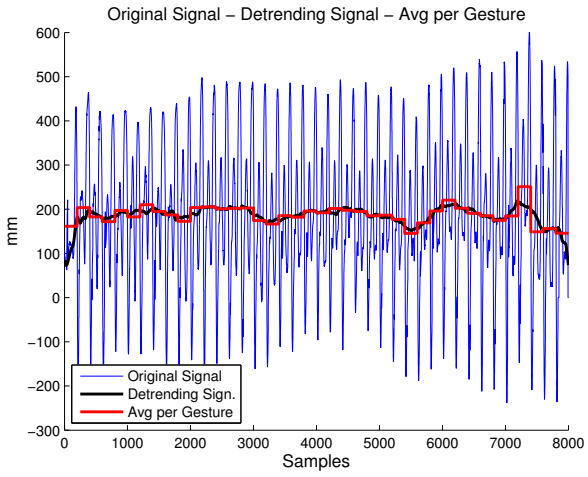


Figure 3. Differences between 2 normalisation methods. Method 1 uses the detrending signal (black) for normalisation. Method 2 uses the average per basic gesture (red) for normalisation .

Our criterion is the maximum log-likelihood of the gesture given the DPGMM (2-3).

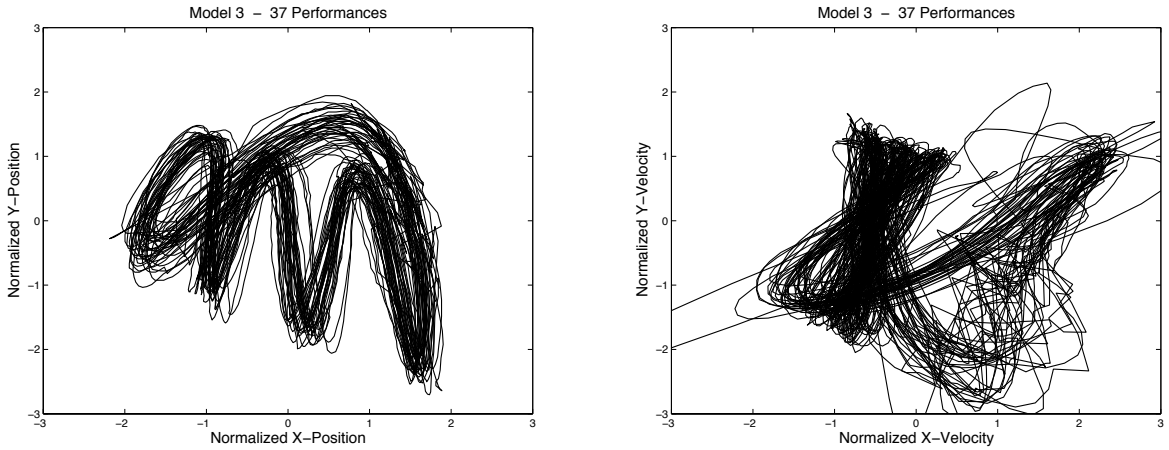
$$p(x|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \quad (2)$$

$$\begin{aligned} m^* &= \arg\max_m(log(P(X_m|\theta))) \\ &= \arg\max_m(\sum_{n=1}^{200} \log(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_{m,n}|\mu_k, \Sigma_k))) \quad (3) \end{aligned}$$

Note that x stands here for a 4D data vector (including position and velocity), K for the number of clusters, π_k for the mixing weight of cluster k and (μ_k,Σ_k) are the mean and the covariance matrix of cluster k. X_m stands for the sequence of x-vectors observed in basic gesture m. The number of states is set equal to the number of segments (vectors with the same winning Gaussian) in the best basic gesture m^*.

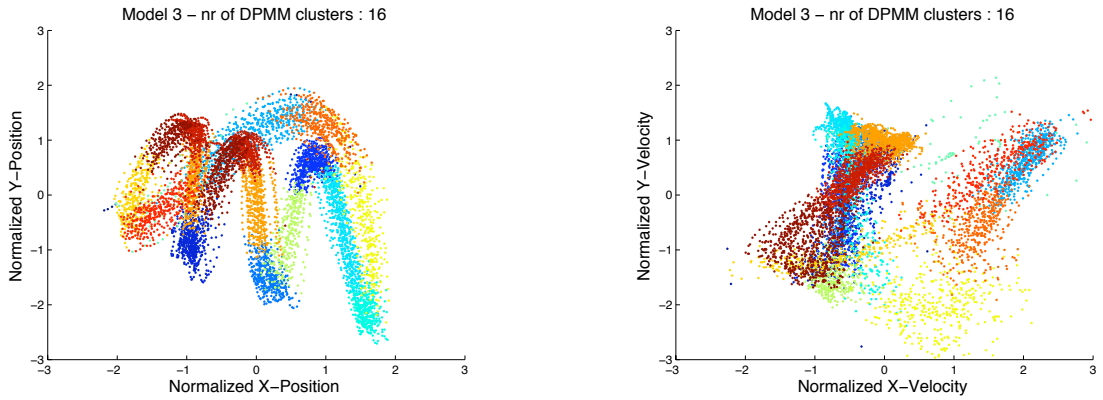
Besides the number of hidden states we need to learn the other HMM parameters as well. HMM parameters are usually denoted as $\lambda = (\pi, A, E)$ the vector of mean vectors and covariance matrices used for computing the emission probabilities.

Without loss of generality we can set $\pi = [1\ 0\ ...\ 0]$ meaning that we always start at hidden state 1. The other parameters (A,E) are learned from the data by means of the Baum-Welch algorithm. The Baum-Welch algorithm needs initial values for the transition probability matrix (A) and for the emission parameters (E). The transition probabilities are set to allow only self-transitions and forward transitions to the next state and to the second next state. Their initial settings are



(a) (b)

Figure 4. Normalized trajectories, with (a) being the positional coordinates, and (b) being the velocity coordinates. The trajectories for 37 basic gestures are shown.



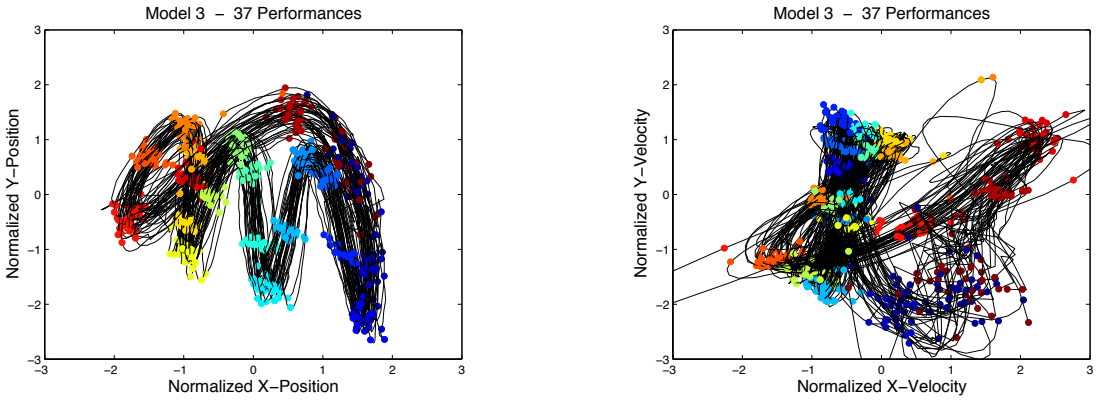
(a) (b)

Figure 5. Cluster assignment (based upon positional coordinates and velocity coordinates).

calculated from the state assignments implied by the best basic gesture m^* found before. Here τ_i represents the duration of the segment corresponding to state i. According to the recommendations of [22] and using Z as a normalizing constant ($\sum_j A_{ij} = 1$), the transition probabilities are set to:

$$\begin{aligned} A_{i,i} &= (1 - \frac{1}{\tau_i})(1/Z) \\ A_{i,i+1} &= \frac{1}{\tau_i}(1/Z) \\ A_{i,i+2} &= \frac{1}{4\tau_i}(1/Z) \end{aligned} \tag{4}$$

As explained before, every cluster (segment) of m^* corresponds with one hidden state. In the initial emission structure we store the mean vector and the covariance matrix of the corresponding Gaussian cluster. All initial parameters are set now and the HMM is ready for training using the Baum-Welch algorithm. Once the HMM is trained, an obvious solution would be to select the basic gesture with the highest log-likelihood given the HMM [23]. This straightforward solution might look attractive at first sight but it fails to handle the temporal variation in an appropriate way. This is because HMMs exhibit some degree of invariance to local warping of the time-axis [24]. We propose to calculate and to define for every basic gesture the most likely hidden state sequence using the Viterbi algorithm and to define the HMM key points where the hidden state transitions occur. However, as shown in Fig. 6, they suffer from positional and temporal variation and in order to solve that problem we apply DTW.



(a) (b)

Figure 6. HMM key points for all basic gestures and their positional variation.

3.4. DTW

Our DTW approach consists out of two steps. In the first step we calculate a reference signal and in the second step we align every basic gesture with that reference signal.

This approach is basically the one advocated in [22] but it is adapted to accommodate the temporal constraint that all gestures complete in one measure. Firstly, we calculate from all basic gestures the average duration of every hidden state. This information is used to subdivide the period of one measure and results in a set of time markers (a time vector) with one marker for every hidden state transition. Now, we use the basic gesture with the highest log-likelihood given the HMM and align this gesture by mapping the timestamps of its key points to the previously produced time vector and this by linear temporal interpolation. This means linear stretching or shrinking of the corresponding state intervals. The resulting signal is the reference signal used for DTW.

Next, we warp all other basic gestures to this reference signal. DTW is preferred here over linear temporal interpolation as it handles the spatial distortion of the signals more efficiently [22].

The DTW procedure requires for every basic gesture a (dis)similarity matrix (D). The task of DTW is to find herein an optimal path. Every element of the dissimilarity matrix ($D_{i,j}$) is calculated as the Euclidean l_2-norm between a 4D sample i of the reference signal s_{ref} and a 4D sample j of the basic gesture s_{bas} (see Eq. 5). Note that some authors recommend a shape preserving time constraint while calculating the optimal path [25].

$$D_{i,j} = \|s_{ref}(i) - s_{bas}(j)\|_2 \tag{5}$$

The similarity matrix is then used to find the sequence of pairs (i, j) forming a path along which the sum of distances $D(i, j)$ is minimal. This path represents a time warping. A comparison of the original gestures and the time warped gestures is displayed in Fig. 7. We define DTW key points as the motion vectors of the warped signals at the previously defined time vector. It is clear that the bundles from the warped signals are more compact, confirming that our procedure takes some of the variance away. The actual DTW implementation was done using a Matlab program by Ellis [26].

3.5. Generalised trajectory

As a result of DTW we have a set of time-warped basic gestures and we have their values (DTW key points) at the newly created time vector. The time vector defines the (non-equidistant) knots for cubic spline regression and the DTW key points are input to the regression. The whole procedure is visualised in Fig. 8. Here the non-equidistant knots (time vector) are symbolised by a red line and the DTW key points (values) are represented by blue dots. The resulting regression line is shown in black.

The regression lines for all coordinates make up the generalised trajectory of a beating-time gesture and form the target trajectory for a robot. This is presented for model 3 in Fig. 9. The red dots correspond here with the calculated time vector used for the DTW key points. The generalised trajectories of the other models can be found in Fig. 10. The trajectories displayed are the targets for the positional coordinates. The targets for the velocity coordinates are not shown here.

3.6. Benchmarking

We benchmarked the results of our method against two other methods. As a first alternative method, we

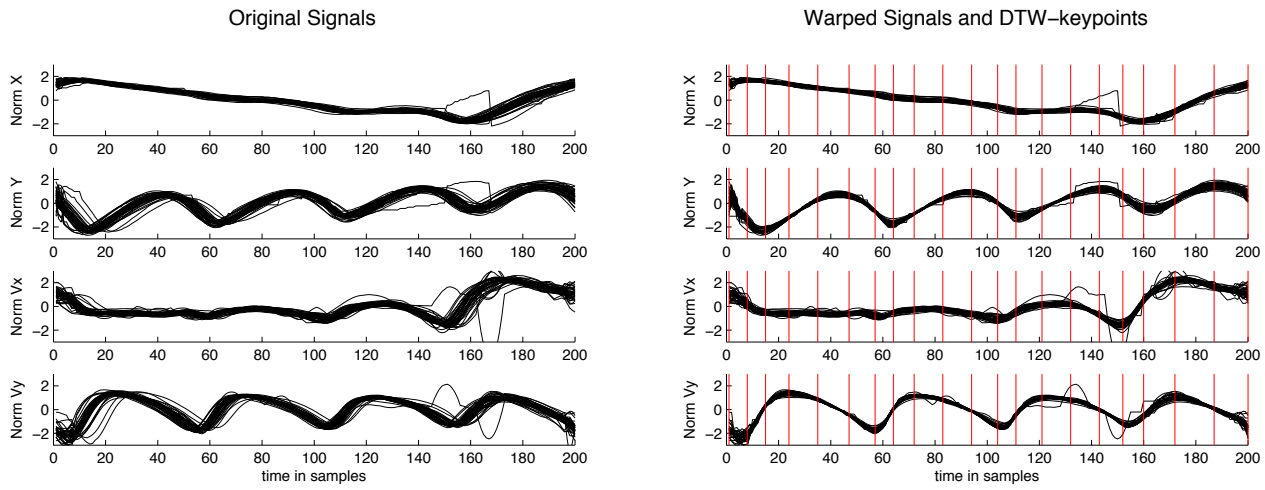


Figure 7. The original signals are shown in the left figure. The warped signals together with the set of time markers (red lines) are shown in the right figure (for conducting model 3).

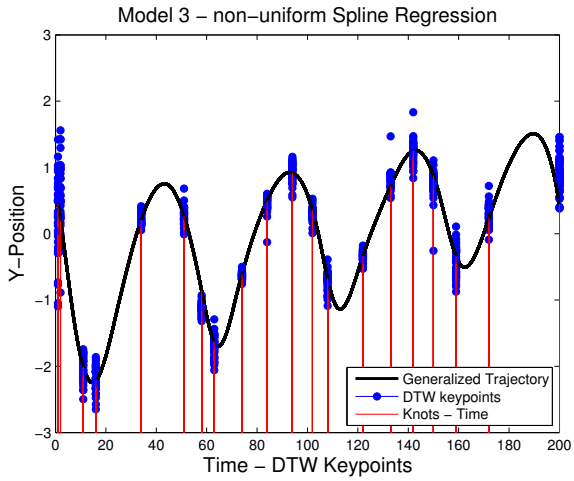


Figure 8. Spline Regression: knots are set at the calculated time vector. The DTW key points are the values from the warped basic gestures at this time vector. The knots and the DTW key points are input to cubic spline regression. The example shown here is for the y-coordinate and is for the generalised trajectory of subject 1 - model 3.

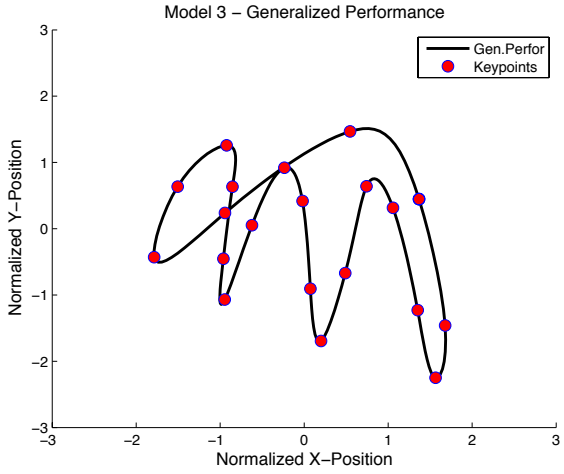


Figure 9. Generalized trajectory and its key points for conducting model 3.

produced a generalised trajectory directly from all basic gestures in the demonstration. Hereby *uniform* cubic splines (having equidistant knots) were used. As a second alternative method, we used Gaussian Mixture Regression (GMR) [27][28]. We set the number of Gaussian components in this method equal to the number of components discovered by our DPGMM. Eventually, we compared our proposed solution of a key point-based generalised trajectory with the two other methods in Fig. 11.

The main difference is that the extrema are more pronounced for our key point-based method compared to the two other methods. The preservation of the extrema is due to the removal of the temporal variance by using DTW. This step should therefore be part of best practice [6].

A next logical question is whether we can define quantitative performance indicators to benchmark these various solutions. This proves to be a difficult point. In literature we find often the Root Mean Square Error (RMSE) as metric for benchmarking [6, 22]. RMSE

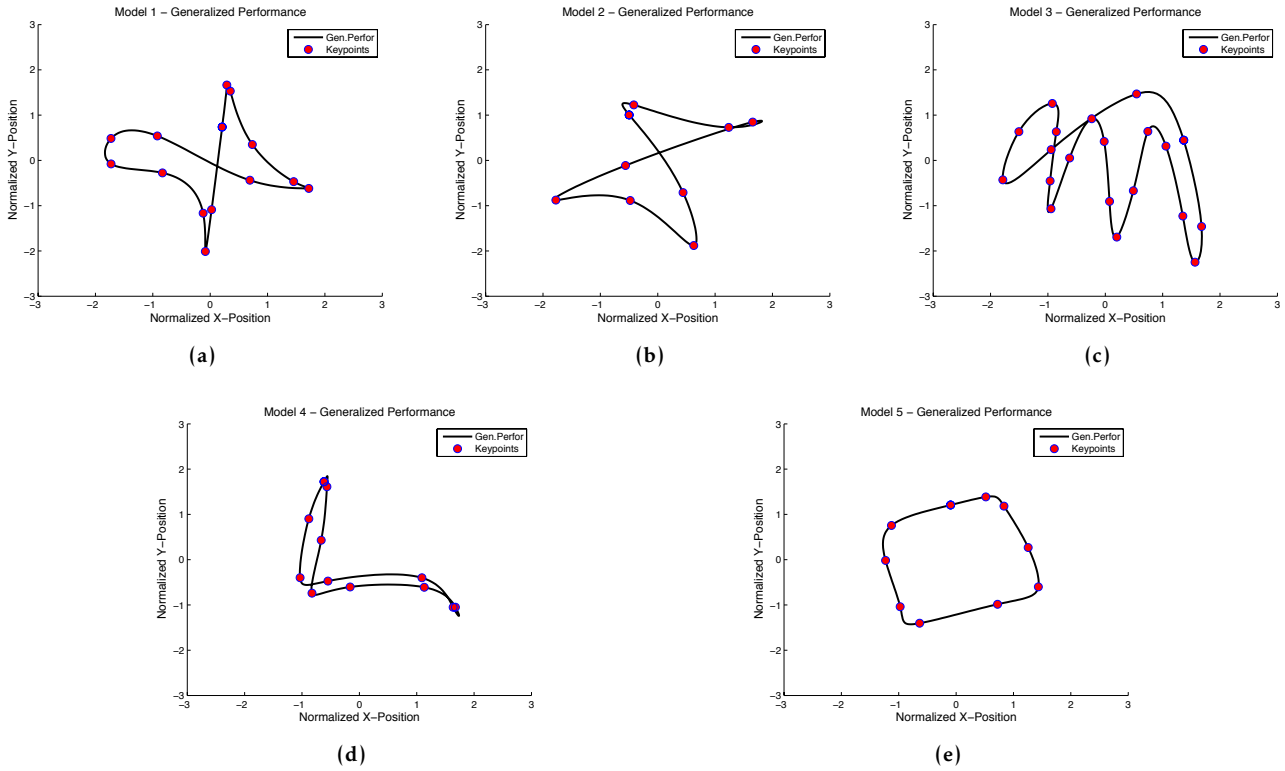


Figure 10. Generalized trajectory and key points of all conducting models.

evaluates how well a gesture x matches another gesture y using Eq. 6.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left\| x(i) - y(i) \right\|^2}{N}} \tag{6}$$

The idea is to use the sum of all RMSE values coming from the comparison of every basic gesture with the generalised trajectory. The major concern with this metric is that it overlooks the temporal variation and that makes it inappropriate for our application. An improvement, namely time warped RMSE, calculates the RMSE values not for the original but for the time warped basic gestures. Although this handles temporal variation there is now the additional issue of what signal should be used as reference for DTW alignment. Selecting one or another reference strongly biases the RMSE results, making this solution also inappropriate. An interesting alternative solution is not to compare the gestures for every timestamp but to compare the curves as a whole. This can be done by defining the area between the curves as distance measurement. This area can be approximated by for example regular resampling. Still, this gives us no measure to express the temporal variation. All these issues made us consider other performance indicators that relate well to the ultimate application. Beating-time gestures use the extrema to convey beat information, so preserving the extrema and their timing is an important performance indicator. Other indicators we propose measure how suitable a target trajectory is for a robot. Candidate indicators are the jerk (derivative of acceleration) of robot movement and also the required on-line computation time.

4. An application

Beating-time gestures indicate the musical beat, meaning that these gestures have temporal targets, or goal points. Godøy defines goal points as certain salient events in the music such as downbeats, or various accent types, or melodic peaks to which sound-producing and sound-accompanying movements are aimed [29]. Goal points link gestures with time and for beating-time gestures, the goal points of interest are the beat times. Note that the goal points are different from the previously discussed key points. Goal points relate to time, whereas key points reflect the shape of a trajectory. The concept of goal points is useful for our application where we want to generate a sequence of beating-time gestures that fit to music. Fitting to music basically means adapting generalised trajectories in terms of musical tempo and musical amplitude.

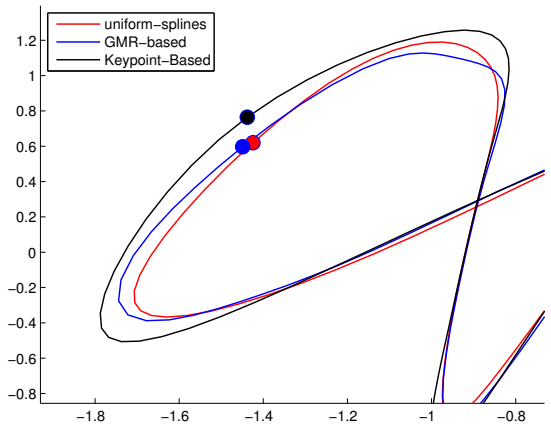


Figure 11. Benchmark of our proposed solution (keypoint-based) against an uniform-splines solution and a GMR solution for model 3. For the uniform-splines solution the knots are set equidistant, this opposed to our Key point-based solution. The GMR solution uses the same number of Gaussian clusters as discovered by our DPGMM method. For visibility reasons this figure zooms in on the top left part of the gesture. For convenience of the reader we added solid circles to all solutions indicating the position of the trajectories at the fourth beat. We notice that the key point-based solution excels in handling the temporal variation as it is better in preserving the extrema.

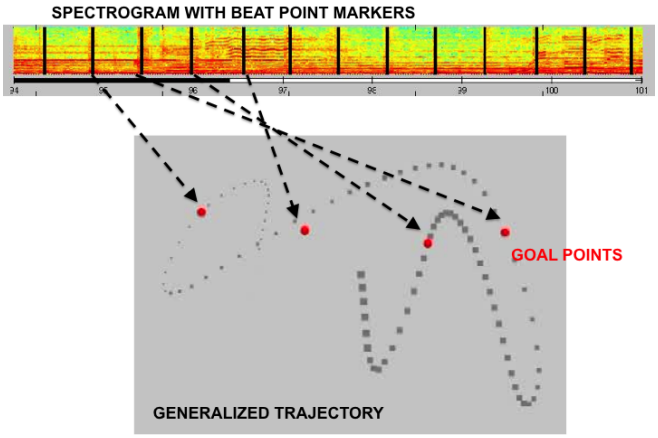


Figure 12. An application: A beating-time gesture for music is made by mapping the goal points (beat times) of a generalised trajectory to the beat points of the music. This is achieved by linear temporal interpolation. The time progress bar shows the actual time stamp of the music and the actual position of the gesture (between beat two and three). This operation changes the run-through speed of the generalised gesture. Additionally the amplitude of the generalised gesture can be changed in accordance with the musical amplitude.

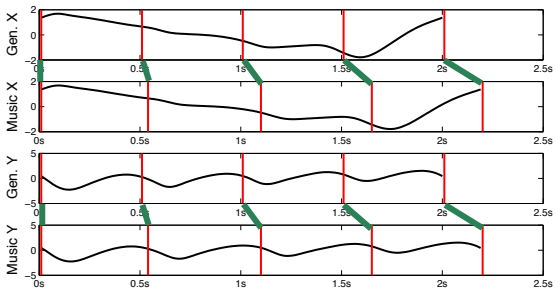


Figure 13. Making a beating-time gesture for music (Music X,Music Y) from a generalised trajectory (Gen X,Gen Y). The general trajectory is labeled Gen X for its X-coordinate and Gen Y for its Y coordinate, the synthesised gesture Music X and Music Y. All horizontal-axes express time in seconds. The procedure maps the goal points (metronomic ticks) of the generalised trajectory onto the goal points (beat points) of the music. The goal points are visualised by vertical red lines, the mapping by green lines. The synthesised beating-time gesture (Music X,Music Y) is calculated by linear temporal interpolation.

For the metronome stimuli used in our experimental set-up (see Section 2), goal points could be easily identified as they coincided with the timestamps of the metronome ticks. For music, the goal points must coincide with the beat points in the music. Beat points in music are typically extracted using some beat tracker program (check McKinney et al. [30] for an overview of beat tracker programs).

To adapt a generalised gesture to music we map the intervals between the goal points from the generalised gesture to the intervals made up by the beat points of the music (Fig. 12, see also a video example at `https://www.youtube.com/watch?v=x521tGb9n1Q&feature=youtu.be`). This can be achieved by stretching and shrinking the time intervals and the easiest way to achieve this is by linear temporal interpolation as is shown in Fig. 13. This works well for positional data but for velocity data an additional step is required. Remember from Eq. 1 that a robot needs a target for position and velocity. For velocity data we do linear temporal interpolation as well but in addition all velocity values have to be changed proportionally to the stretch of the time interval. If the time interval doubles (i.e., music has a slower tempo than the metronome), the velocity should be set to half. We recall that the generalised trajectory for our conducting gesture is made from a set of normalised performances. That makes the generalised trajectory also normalised and easily scalable. Scales can be chosen in accordance with musical amplitude.

5. Discussion

The main aim of this study was to provide a PbD solution to the generation of generalised trajectories of periodic, multi-segmented beating-time gestures - on the basis of a (small) set of demonstrations. In essence, our solution involved a dynamic time warping approach that focused on the construction of a reference signal for time warping. The reference signal holds temporal information coming from all basic gestures and is calculated from the hidden state transitions of a fitted HMM. This calculation involves an averaging step and as such it is outlier sensitive. Hence, care should be taken to remove outlying basic gestures prior to the analysis. To identify outliers or even to inspect the quality of a performance we recommend to use a visual tool like for example "Gesture Heatmaps" [31]. With "Gesture Heatmaps" any localized feature can be plotted against the gesture paths helping to identify areas of difficulty along the path.

The whole procedure has quite some similarities to methods used for speech recognition and speech synthesis. In that regard, it follows the idea of a vocoder being an analysis/synthesis system, used to reproduce human speech. For modelling speech, HMMs are the de-facto standard. For synthesis however, an HMM does not perform well since the duration model (hidden state self transitions) is rather simplistic [32]. To attain good performance, a separate duration model is utilised to fix the instances were the state transitions have to occur. Such a system is actually no longer an HMM system. It is called a Hidden Semi-Markov Model (HSMM).

The location of the goal points for our calculated generalised trajectories contradicts the intuitive understanding of a conducting gesture that most of us have. Most people anticipate the beats to occur at the extremities of the conducting gesture movement. For example for model 1 this is at the top, bottom, left and right position. Our research learns however that there is a lag of approximately 0.25s (compared to the 2s for the bar) between these positions and the actual beat points. Although our study was limited to four subjects and generalisation is impossible, this result is in line with a previous study from Luck and Toiviainen [33]. In their study, the authors found that an ensemble's performance, executed in an ecological setting with a conductor, tended to be most highly synchronised with periods of maximal deceleration along the trajectory, in second place followed by periods of high vertical velocity.

A critic on our method could be that the synthesised beating-time gesture is not human. During the production process of a generalised trajectory we focused on timing, leading to an artificial trajectory rather than on the human factor, what would mean selecting one performance out of a set. However, our artificial gesture was eventually *humanised* by making it smooth through cubic spline regression. In addition, our system is still off-line: We extract the beat points off-line and up-front and we use them to generate a synthesised beating-time gesture also off-line and up-front. Moving from an off-line beat detection algorithm to an on-line beat detection algorithm would make it possible for a conductor to adapt the timing of his gestures to what the orchestra is actual playing. We suggest to follow an adaptive learning approach, based on a maximum a posteriori (MAP) estimation, and integrating the propagated knowledge from previous time intervals.

Our work is an initial but important step towards a fully automated conducting system. Such a system, either in the form of a robot or animated avatar, may help people in performing beating-time gestures in response to music. Consequently, this may reinforce people's ability to better discriminate and understand temporal structures of music. Additionally, robots or animated avatars may be attractive to people and stimulate them to be active and involved, which is a major asset in music-pedagogical settings. Our present implementation is now limited to beating-time gestures. A next step could be to move to a more extended set of gestures, such as dance movements. Also the procedures may be implemented in systems for motor rehabilitation purposes.

6. Conclusion

In this paper, we have developed a Programming by Demonstration (PbD) method for generating a generalised trajectory from a set of demonstrated periodic, multi-segmented beating-time gestures. Beating-time gestures are peculiar in the sense that they are targeted at specific moments in time - indicated by musical beats - while performing a particular spatial pattern. Therefore, a major asset of our method is its ability to cope with both the spatial and temporal variation within a set of demonstrated periodic gestures. To that end, it utilises two probabilistic models, namely a DPGMM and a HMM, together with a DTW algorithm. A secondary achievement of the paper is the development of a procedure to adapt a generalised conducting pattern to music on the basis of so-called goal points (temporal targets). In future research, these procedures may be used to drive robots or animate avatars in contexts of music, dance, or motor rehabilitation.

Acknowledgment

This research was conducted in the framework of the EmcoMetecca project, granted by Ghent University - Methusalem-BOF council. We want to thank Ivan Schepers for the technical assistance.

References

[1] Schaal S. Is imitation learning the route to humanoid robots? Trends in Cognitive Sciences. 1999;3(6):233–242.

[2] Billard A, Calinon S, Dillmann R, Schaal S. Robot programming by demonstration. In: Siciliano B, Khatib O, editors. Springer handbook of robotics. Berlin, Germany: Springer; 2008. p. 1371–1394.

[3] Wilson AD, Bobick AF. Hidden Markov models for modeling and recognizing gesture under variation. International Journal of Pattern Recognition and Artificial Intelligence. 2001;15(1):123–160.

[4] Ude A. Trajectory generation from noisy positions of object features for teaching robot paths. Robotics and Autonomous Systems. 1993;11(2):113–127.

[5] Aleotti J, Caselli S. Robust trajectory learning and approximation for robot programming by demonstration. Robotics and Autonomous Systems. 2006;54(5):409–413.

[6] Calinon S, D'halluin F, Sauser EL, Caldwell DG, Billard AG. Learning and reproduction of gestures by imitation. IEEE Robotics & Automation Magazine. 2010;17(2):44–54.

[7] Sprunk C, Lau B, Burgard W. Improved non-linear spline fitting for teaching trajectories to mobile robots. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2012. p. 2068–2073.

[8] Naveda L, Leman M. The spatiotemporal representation of dance and music gestures using topological gesture analysis (TGA). Music Perception. 2010;28(1):93–111.

[9] Haugen MR, Godøy RI. Rhythmical structures in music and body movement in Samba performance. In: Proc. of the ICMPC-APSCOM 2014 Joint Conference. Society for the Cognitive Sciences of Music; 2014. p. 46–52.

[10] Fan R, Xu S, Geng W. Example-based automatic music-driven conventional dance motion synthesis. IEEE Transactions on Visualization and Computer Graphics. 2012;18(3):501–515.

[11] Ofli F, Erzin E, Yemez Y, Tekalp AM. Learn2dance: Learning statistical music-to-dance mappings for choreography synthesis. IEEE Transactions on Multimedia. 2012;14(3):747–759.

[12] Oliveira JL, Naveda L, Gouyon F, Reis LP, Sousa P, Leman M. A parameterizable spatiotemporal representation of popular dance styles for humanoid dancing characters. EURASIP Journal on Audio, Speech, and Music Processing. 2012;2012(1):1–20.

[13] Okamoto T, Shiratori T, Kudoh S, Nakaoka S, Ikeuchi K. Toward a dancing robot with listening capability: Keypose-based integration of lower-, middle-, and upper-body motions for varying music tempos. IEEE Transactions on Robotics. 2014;30(3):771–778.

[14] Bertini G, Carosi P. Light baton system: A system for conducting computer music performance. Journal of New Music Research. 1993;22(3):243–257.

[15] Lee E, Karrer T, Borchers J. Toward a framework for interactive systems to conduct digital audio and video streams. Computer Music Journal. 2006;30(1):21–36.

[16] Bevilacqua F, Guédy F, Schnell N, Fléty E, Leroy N. Wireless sensor interface and gesture-follower for music pedagogy. In: International Conference on New Interfaces for Musical Expression (NIME). ACM; 2007. p. 124–129.

[17] Johannsen G, Nakra TM. Conductors' gestures and their mapping to sound synthesis. In: Musical gestures: Sound, movement, and meaning. New York, NY: Routledge; 2010. p. 264–298.

[18] Phillips-Silver J, Trainor LJ. Feeling the beat: movement influences infant rhythm perception. Science. 2005;308(5727):1430–1430.

[19] Phillips-Silver J, Trainor LJ. Hearing what the body feels: Auditory encoding of rhythmic movement. Cognition. 2007;105(3):533–546.

[20] Teh YW. Dirichlet Process. In: Sammut C, Webb GI, editors. Encyclopedia of machine learning. Berlin, Germany: Springer; 2007. .

[21] El-Arini K. Dirichlet Processes : a gentle tutorial; 2008.

[22] Vakanski A, Mantegh I, Irish A, Janabi-Sharifi F. Trajectory learning for robot programming by demonstration using hidden Markov model and dynamic time warping. IEEE Transactions on Systems, Man, and Cybernetics, Part B. 2012;42(4):1039–1052.

[23] Tso SK, Liu KP. Demonstrated trajectory selection by hidden Markov model. In: Proc. of the IEEE International Conference on Robotics and Automation. vol. 3. IEEE; 1997. p. 2713–2718.

[24] Bishop CM. Pattern recognition and machine learning. Berlin, Germany: Springer; 2006.

[25] Sakoe H, Chiba S. Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing. 1978;26(1):43–49.

[26] Ellis D. Dynamic time warp (DTW) in Matlab;. Available from: http://www.ee.columbia.edu/~dpwe/resources/matlab/dtw/ [cited May 19, 2017].

[27] Calinon S, F G, A B. On learning, representing and generalizing a task in a humanoid robot. IEEE Transactions on Systems, Man and Cybernetics, Part B. 2007;37(2):286–298.

[28] Cohn DA, Ghahramani Z, Jordan MI. Active learning with statistical models. In: Advances in neural information processing systems; 1995. p. 705–712.

[29] Godøy RI, Jensenius AR, Nymoen K. Production and perception of goal-points and coarticulations in music. Journal of Acoustical Society of America. 2008;123:3657.

[30] McKinney MF, Moelants D, Davies MEP, Klapuri A. Evaluation of audio beat tracking and music tempo extraction algorithms. Journal of New Music Research. 2007;36(1):1–16.

[31] Vatavu RD, Anthony L, Wobbrock JO. Gesture heatmaps: Understanding gesture performance with colorful visualizations. In: Proceedings of the 16th International Conference on Multimodal Interaction. ACM; 2014. p. 172–179.

[32] King S. An introduction to statistical parametric speech synthesis. Sadhana. 2011;36(5):837–852.

[33] Luck G, Toiviainen P. Ensemble musicians' synchronization with conductors' gestures: An automated feature-extraction analysis. Music Perception. 2006;24(2):189–200.