

A Generative Model for Motion Synthesis and Blending Using Probability Density Estimation

Dumebi Okwechime and Richard Bowden

University of Surrey, Guildford, Surrey, GU2 7XH, UK
{d.okwechime,r.bowden}@Surrey.ac.uk

Abstract. The main focus of this paper is to present a method of reusing motion captured data by learning a generative model of motion. The model allows synthesis and blending of cyclic motion, whilst providing it with the style and realism present in the original data. This is achieved by projecting the data into a lower dimensional space and learning a multivariate probability distribution of the motion sequences. Functioning as a generative model, the probability density estimation is used to produce novel motions from the model and gradient based optimisation used to generate the final animation. Results show plausible motion generation and lifelike blends between different actions.

Keywords: Animation, Motion capture, Probability Distribution Function, Principal Component Analysis, Least Square Curve Fitting, Linear Interpolation.

1 Introduction

The purpose of this work is to develop a generative model of motion that allows motion synthesis and blending of different cyclic movements such as walking or running. The model is capable of generating novel motion whilst preserving the same style and realism of the original motion capture data. This allows animators to edit, extend and blend between different cyclic movements providing the ability to reuse motion captured data. This paper builds upon previous work into synthesising motion using probability density estimation and proposes a multivariate probability distribution for use in synthesis and blending.

The human visual system has the ability to efficiently and easily recognise characteristic human movement. As a consequence, in order to generate character animations that look realistic, it is necessary to develop methods to capture, maintain and synthesis intrinsic style to give authentic realism to animated characters. The aim of this work is to provide the general framework that addresses this issue. The framework is based on transforming biological motion into a representation that allows analysis using statistical techniques. Additionally, this model will also be used to synthesise and blend realistic motion patterns.

In this paper, we present two approaches to motion synthesis and blending. One approach uses a sine function to model a cyclic motion sequence and utilises linear interpolation to blend between different motions. The second, more flexible

approach, models motion as a probability density function (PDF). This model can synthesise novel motion whilst retaining the natural variances inherent in the original data. Blending is a result of linearly interpolating between different PDF's.

This paper is divided into the following sections. Section 2, briefly details current techniques of modelling and editing motion capture data. Section 3 and Section 4 present the techniques used for learning models that capture the statistics and dynamics of a set of motion capture sequences. The remainder of the paper describes the results and ends with a brief conclusion.

2 Related Work

Character animation using interpolation was one of the earliest techniques used in computer animation and is still used by many animators today [1]. The traditional technique uses interpolation to generate intermediate frames from manually defined key-frames [2]. This method offers high controllability to the animator, but it is very time consuming and requires a highly skilled animator to produce the animation.

Motion capture provides a cost effective solution to realism as life-like motion is easily acquired and large libraries of motion are available for use. However, to provide realistic results, multiple sequences need to be blended together resulting in a seamless and life-like animation. Furthermore, recorded animation lacks the variability of natural motion and repeating a motion can look false due to its lack of natural variance.

One solution is to develop a generative model of motion using motion capture data. With the ability to blend between motions, one can create a sequence of realistic actions which would otherwise be difficult to achieve using keyframing. This approach is made possible by fitting a parametric function to the joint coordinates of the character and interpolating between discrete points in this parametric space. An early approach was presented by Rose et al. [3]. They defined each degree of freedom of a character as a uniform cubic B-spline curve in time. With this representation, they use radial basis functions (RBF) to interpolate between sample motions in this space. Kovar et al. [4] improved on this approach by introducing motion graphs which automatically construct novel motion from a mass of motion capture data. By representing frames in terms of point clouds, they calculate the weighted sum of squared distance between corresponding points in the clouds. If the distance is below a user-specified threshold, the relative motions are considered similar. Similar motions can then be blended by linearly interpolating the corresponding root positions and spherically linearly interpolating their corresponding joint rotations.

Carvalho et al. [5] follows a similar approach to this paper using Principal Component Analysis (PCA) to learn motion models from motion captured data. However, they use a Prioritised Inverse Kinematics strategy to apply constraints with different levels of importance to synthesis the motion. As opposed to using PCA, Grochow et al. [6] learns a PDF over character poses represented by a

Scaled Gaussian Process Latent Variable Model. This model represents the data in a low-dimensional latent space, and motion synthesis occurs by optimising the likelihood of new poses given the original poses.

Unuma et al. [7] create a functional model of motion using Fourier series expansion. The model is used to make variations of human behaviour via the interpolation and extrapolation of their Fourier coefficients. In related work, Troje [8] uses sine functions instead of Fourier expansion to model walks. Using PCA, they extract relevant information from the data, representing them as discrete components. The temporal behaviour of these components is modelled using a sine function, and sinusoidal curve fitting is used to parameterise them based on their respective frequency, amplitude and phase. A limitation to this approach is that it produces identical motion cycles which are not natural, and the sine assumption limits the type of motion that can be modelled.

Pullen and Bregler [9] introduce the idea of synthesising motion by extracting the 'motion texture' i.e. the personality and realism, from the motion and using it to drive hard constraints such as foot positions on the floor. They comprise their motion of three important features, frequency band, phase, and correlation. These features are represented with a kernel-based probability distribution. This distribution is used to synthesis the walk and a gradient based method used to optimise the data. Part of this approach is adopted in the paper whereby we use a multivariate probability distribution to model the data and synthesis a walk, however, we are also able to blend between different distributions to create novel motion.

3 Methods

3.1 Walking Data

The motion captured data used in this work are in a format which details the 3D Cartesian coordinates (x,y,z)[10] for all the markers corresponding to the frames for the full body motion capture data, although similar approaches can be applied in polar spaces. Given a motion sequence \mathbf{X} , each frame is formally represented as a vector \mathbf{x}_i where $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_F]$ and F is the number of frames containing R markers. Each frame/posture is represented as a high dimensional vector $\mathbf{x}_i = [x_{i1}, y_{i1}, z_{i1}, \dots, x_{iR}, y_{iR}, z_{iR}] \in \mathfrak{R}^{D=3R}$.

3.2 Dimensional Reduction and Statistical Modelling

Working with full-body, motion capture data produces a high dimensional dataset. However, the dimensionality of the resulting space does not necessarily reflect the true dimensionality of the subspace that the data occupies. Principal Component Analysis (PCA) [11] is therefore used to perform dimensional reduction.

For a given D -dimensional data set \mathbf{X} as defined in Section 3.1, the D principal axes $\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_D$ can be given by the D leading eigenvectors of the sample covariance matrix:

$$\mathbf{S} = \frac{1}{N} \sum_{i=1}^F (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \tag{1}$$

where $\boldsymbol{\mu}$ is the sample mean $\boldsymbol{\mu} = \frac{1}{F} \sum_{i=1}^F \mathbf{x}_i$. An eigen decomposition gives $\mathbf{S} = \sum \lambda_i \mathbf{T}_i$, $i \in \{1, \dots, D\}$, where λ_i is the i th largest eigenvalue of \mathbf{S} .

The dimension of the data can be reduced by projecting into the eigenspace

$$\mathbf{y}_i = \mathbf{V}^T (\mathbf{x}_i - \boldsymbol{\mu}) \tag{2}$$

where \mathbf{V} are the feature vectors $\mathbf{V} = [\mathbf{T}_1, \dots, \mathbf{T}_d]$ and d is the chosen lower dimension $d \leq D$. This results in a d -dimensional representation of each motion sequence in eigenspace. We will refer to each dimension as modes of variation within the data. d is chosen such that $\sum_{i=1}^d \frac{\lambda_i}{\sum \lambda} \geq .95$ or 95% of the energy is retained.

Figure 1 shows a plot of the first mode against the second mode for a male and female walk sequence. It produces a geometric shape characteristic of a cyclic motion, showing that the projection retains the non-linearity of cyclic movement.

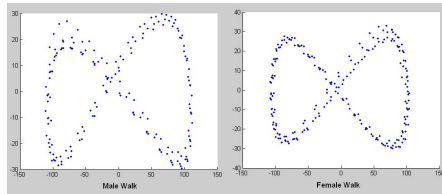


Fig. 1. Plot of PCA projection of the first mode against the second mode

3.3 Time Alignment

Using the PCA space data representation detailed in Section 3.2, the first mode of each motion sequence can be represented as discrete sine waves, a technique developed by Troje [8]. This is possible since walks are cyclic movements that repeat sinusoidally. Using least square curve fitting, the first mode of each motion sequence is parameterised as:

$$\sin(\omega t + \varphi) \tag{3}$$

where ω is the fundamental frequency and φ is the phase.

It can be seen in Figure 2 that the first mode is almost a perfect sine wave and is closely approximated by the sine function, the parameters of which can be used to normalise the data.

$$t' = ((t * \omega) + \varphi) \tag{4}$$

where t' is the new time.

Since the first mode contains most of the motion’s variance, it is not necessary to fit a sine function to the subsequent modes. The first mode is sufficient to normalised the time domain of the sequence.

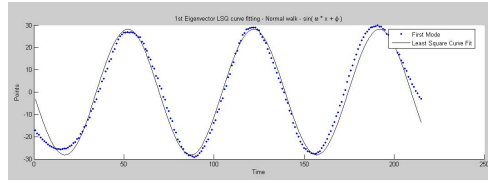


Fig. 2. Least square curve fitting of first mode wave representation for a male walker

Blending using Time Aligned Sine Waves. With two or more walking sequences time aligned as above, it is possible to blend between them using linear interpolation. For two sequences, male walk (mw) and female walk (fw), blending can be achieved using linear interpolation, where a weight w indicates the contribution of each sequence to the resulting motion.

$$\mathbf{y}'(t) = \frac{\mathbf{y}_{mw}(t) + \mathbf{y}_{fw}(t)}{2} + w \frac{\mathbf{y}_{mw}(t) - \mathbf{y}_{fw}(t)}{2} \tag{5}$$

where \mathbf{y} is the respective posture at a given time t , and w is the weighting variable ($-1 < w < 1$) where $w = 1$ for the male walk animation, $w = -1$ for the female walk animation and $w = 0$ for the mid-point between them.

However, there are a number of issues with this approach. Firstly, time alignment will only work well if the motion is well represented by a sine wave. Secondly, it can only blend between similar cyclic motions such as walks or runs and cannot provide successful blends between more sophisticated motions. It also destroys the time/velocity information during normalisation. Finally, the synthesised motions have no variability, unless the original data is retained and resampled. In the next section we overcome these problems by representing the original data as a PDF in the eigenspace.

4 Generative Model of Motion Using a Probability Distribution Function

A statistical model of the constraints and dynamics present within the data can be created using a probability distribution. This model is created using kernel estimation. Each kernel $p(\mathbf{y}_i)$ is effectively a Gaussian centred on a data example. Since we want our probability distribution to represent the dimensionally reduced data set Y of d dimensions, where $Y = \{\mathbf{y}_i\}_{i=1}^F$ as noted in Section 3.2, the likelihood of a pose in eigenspace is modelled as a mixture of Gaussians using multivariate normal distributions.

$$P(\mathbf{y}) = \frac{1}{F} \sum_{i=1}^F p(\mathbf{y}_i) \tag{6}$$

Figure 3 shows a plot of such a distribution for the case of a male and female walker with the first mode plotted against the second mode. The width of the Gaussian in the i^{th} dimension is set to $\alpha\sqrt{\lambda_i}$. For these experiments $\alpha = 0.25$.

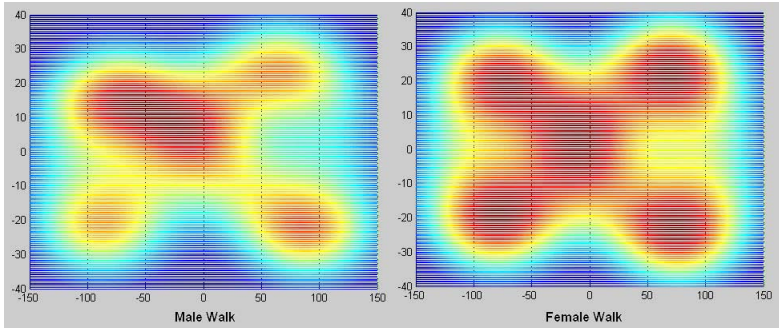


Fig. 3. PDF of a male walk sequence vs female walk sequence

4.1 Motion Synthesis

To generate novel motion the procedure is:

(1) $P(\mathbf{y})$ is constructed as PDF in the pose space that gives the likelihood of any particular pose configuration.

(2) As we are particularly interested in motion, a second PDF is constructed that encodes the likelihood of motion in the pose space for a given configuration $P(\mathbf{y}, \frac{d\mathbf{y}}{dt})$ where

$$\frac{d\mathbf{y}_i}{dt} = \mathbf{y}_{i+1} - \mathbf{y}_i \tag{7}$$

assuming regular sampling over the motion capture data. $P(\mathbf{y}, \frac{d\mathbf{y}}{dt})$ is constructed similarly to Equation 6 using F Gaussian kernels in \mathbb{R}^{2d} . Similarly to Equation 6, the covariance is set to

$$\alpha \begin{pmatrix} \sqrt{\lambda_1} & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \sqrt{\lambda_d} & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \sigma_1 & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \ddots & \cdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \sigma_d \end{pmatrix} \tag{8}$$

where σ_i is the standard deviation of the derivatives.

(3) To locate a suitable starting configuration, the kernel that generates the highest likelihood is found.

$$\max = \arg \max_{i=1}^F (P(\mathbf{y}_i)) \tag{9}$$

(4) From this configuration $\mathbf{y}_t = \mathbf{y}_{\max}$ the highest likelihood movement is selected

$$\max \Delta = \arg \max_{\forall \frac{d\mathbf{y}}{dt}} (P(\mathbf{y}_t, \frac{d\mathbf{y}}{dt})) \tag{10}$$

(5) The model pose is then updated such that

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \frac{d\mathbf{y}_{max\Delta}}{dt} \quad (11)$$

and the pose is then reconstructed for rendering as $\mathbf{x}_{t+1} = \mu + \mathbf{V}\mathbf{y}_{t+1}$

(6) The process then repeats from step (4).

4.2 Correction Term Constraint

To increase the robustness of the PDF distribution and to account for the increasing probability estimation error, a correction term is added to the optimisation process. The correction term constrains the estimated posture to remain within the PDF. This is brought about by adding a weighting to all estimated postures. Each estimation is multiplied by the likelihood of that posture being a valid posture, discouraging movement outside the pose PDF. Step (4) therefore becomes

$$max\Delta = \arg \max_{\forall \frac{d\mathbf{y}}{dt}} (P(\mathbf{y}_t, \frac{d\mathbf{y}}{dt})P(\mathbf{y}_t + \frac{d\mathbf{y}}{dt})) \quad (12)$$

This improves the optimisation in step (4), reducing drift, resulting in plausible poses.

4.3 Gradient Based Optimisation

Gradient based optimisation works well with the assumption that the optimisation in stage (4) finds a good global maximum. Since the surface is smooth (due to the use of Gaussian kernels) and the space contiguous (in that the positions of two adjacent frames are spatially close in the eigenspace), a simple gradient ascent method can be used. A Mean Shift approach works well since it is only necessary to assess the likelihood at the corners of the search region (again assuming the surface is smooth). However, such a search is $O(D^2k)$ where k is the number of iterations to convergence and D , the dimensionality of the space. It is worth noting that in the case of $P(\mathbf{y}, \frac{d\mathbf{y}}{dt})$ this dimensionality is twice the number of eigenvectors retained in the projection. Line optimisation methods such as Powells Method work well as the surface is smooth and the search become linear in the number of dimensions $O(Dk)$. Newton type methods requiring Jacobian and Hessians are more problematic due to the large number of kernels in the PDF. We therefore optimise along the direction of each eigenvector in turn, using the previous location as the starting point of the search.

4.4 Blending Using a PDF

Blending between different sequences using the PDF follows a very similar procedure to motion synthesis. The different walk sequences are firstly projected

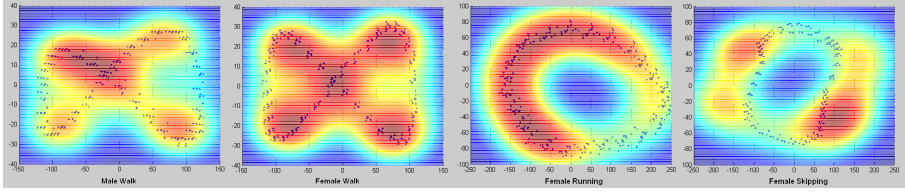


Fig. 4. Plot of four synthesised cyclic motion sequences superimposed over their respective distributions

into the same eigenspace by performing PCA on their combined data (as explained in Section 3.2). After projection down into the combined eigenspace, separate PDF's are constructed for each of the sequences. The PDF used in synthesis is the weighted average of all distributions. By changing the weighting, the influence of each sequence on the final animation can be changed and transitions between the sequences made. If $P(\mathbf{y}_a) = \frac{1}{F_a} \sum_{i=1}^{F_a} P(\mathbf{y}_i^a)$ for walk a and $P(\mathbf{y}_b) = \frac{1}{F_b} \sum_{i=1}^{F_b} P(\mathbf{y}_i^b)$ for walk b , then step (3) is replaced with:

$$max = \arg \max_{i=1}^F \left(\frac{1}{2} (P_a(\mathbf{y}_t) + P_b(\mathbf{y}_t)) + w \frac{1}{2} (P_a(\mathbf{y}_t) - P_b(\mathbf{y}_t)) \right) \quad (13)$$

and similarly, step (4) is replaced with:

$$max \Delta = \arg \max_{\nabla \frac{d\mathbf{y}}{dt}} \left(\frac{1}{2} \left(P_a(\mathbf{y}_t, \frac{d\mathbf{y}}{dt}) + P_b(\mathbf{y}_t, \frac{d\mathbf{y}}{dt}) \right) + w \frac{1}{2} \left(P_a(\mathbf{y}_t, \frac{d\mathbf{y}}{dt}) - P_b(\mathbf{y}_t, \frac{d\mathbf{y}}{dt}) \right) \right) \quad (14)$$

However, with slight rearranging, these equations can be re-written to reduce the number of access to the kernel functions.

5 Results

To illustrate the approach, 4 motion capture sequences were projected down into their combined eigenspace and 4 models constructed each consisting of two PDF's, one for pose and one for motion. The datasets were 'male walk', 'female walk', 'female running', and 'female skipping'. Figure 4 shows a visualisation of the PDFs of 4 sequences projected onto the two primary eigenvectors. It can be seen that these distributions vary considerably but occupy the same space. Using the outlined procedure, a maximum likelihood path was generated from each of the PDF's. The result of which are superimposed on top of their respective distributions. As can be seen, the data points, corresponding to a sequence of postures, remain within the distributions and follow the characteristic shapes of their original data. Also note that small errors in the optimisation procedure produce natural variation in the trajectory. An additional noise term could be added to the estimation procedure, however, in practice these inherent errors are sufficient to produce natural variation 'novelty'.

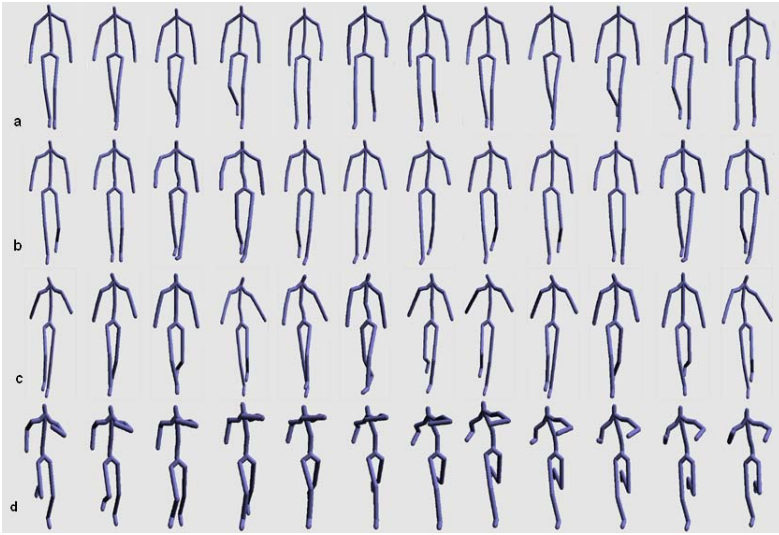


Fig. 5. Image showing animations of synthesised walks using PDF. (a) Male walk, (b) Female walk, (c) Female skipping, (d) Female running.

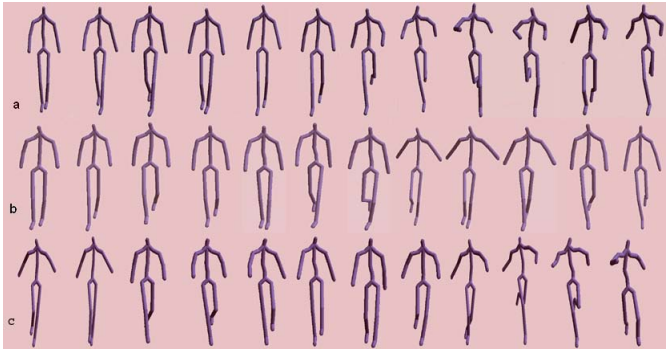


Fig. 6. Image showing animations of blended walks using PDF. (a) Blend from a female walk to a female run, (b) Blend from a male walk to female skipping, (c) Blend from a female skipping to female run.

When animated (as shown in Figure 5), they resemble their original data and produce plausible animations.

Figure 6 shows the effect of blending between motions by changing the weighting attributed to the contribution of any one PDF to the overall density estimate. Again it can be seen that smooth and natural transition are achieved, even when moving between less usual motions such as ‘male walk’ to ‘female skip’.

6 Conclusion

It has been shown that a generative model of motion can be used to synthesis and blend novel motion, whilst retaining the important characteristic motion features from the original data. As the results show, it is possible to learn from motion captured data and provide unlimited novel animation that blends between motions automatically. Style is chosen by selecting the relative contribution of different styles and motions using simple weights. In addition, as no time alignment is required for the process, the temporal information in the animation is preserved. This approach can be extended to blend between different types of non-cyclic movement. The success of this is dependent upon the quantity of motion sequences used, since a larger more disparate eigenspace would require more samples to provide sufficient connectivity between the distributions. Further work will investigate this possibility.

References

1. Tanco, L.M.: Human Motion Synthesis from Captured Data. PhD in Computer vision, University of Surrey, University of Surrey, Guildford, Surrey, GU2 7XH, UK (2002)
2. Ahmed, A.: Parametric Synthesis of Human Motion. PhD in Computer vision, University of Surrey, University of Surrey, Guildford, Surrey, GU2 7XH, UK (2003)
3. Rose, C., Cohen, M., Bodenheimer, B.: Verbs and adverbs: Multidimensional motion interpolation. In: IEEE Computer Graphics and Applications, September/October, pp. 32–40. IEEE Computer Society Press, Los Alamitos (1998)
4. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. In: Proceedings of ACM SIGGRAPH, July, vol. 21(3), pp. 473–482 (2002)
5. Carvalho, S.R., Boulic, R., Thalmann, D.: Interactive low-dimensional human motion synthesis by combining motion models and pik. *Computer Animation and Virtual Worlds* (in press, 2007)
6. Grochow, K., Martin, S.L., Hertzmann, A., Popovic, Z.: Style-based inverse kinematics. In: Proceedings of the 2004 SIGGRAPH Conference, pp. 522–531 (2004)
7. Unuma, M., Anjyo, K., Takeuchi, R.: Fourier principles for emotion-based human figure animation. In: Proceedings of the 22nd annual ACM conference on Computer graphics, Los Angeles, California, August 06–11, pp. 91–96. Addison Wesley, Reading (1995)
8. Troje, N.F.: Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *J. Vis.* 2(5), 371–387 (2002)
9. Pullen, K., Bregler, C.: Synthesis of cyclic motions with texture (2002)
10. Chung, H.S., Lee, Y.: Mcml: motion capture markup language for integration of heterogeneous motion capture data. *Computer Standards and Interfaces* 26, 113–130 (2004)
11. Alexa, M., Muller, W.: Representing animations by principal components. *Computer Graphics Forum*. 19(3), 411–418 (2000)