Scaled Functional Principal Component Analysis for Human Motion Synthesis

Han Du[1] * Somayeh Hosseini[1] Martin Manns[2] Erik Herrmann[1] Klaus Fischer[1]

[1] German Research Center for Artificial Intelligence, Saarbrücken, Germany
[2] Daimler AG, Ulm, Germany

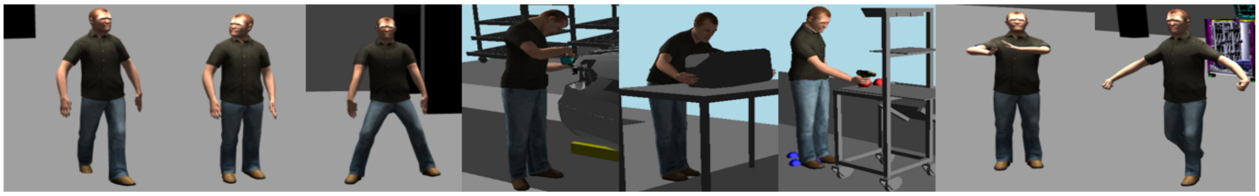

Figure 1: *Generated motion clips from our statistical motion database. From left to right: walking leftStance, looking around, walking sidestep, screwing, two-hand placing, right-hand picking, two-hand transferring and two-hand carrying.*

Abstract

Many of the existing data-driven human motion synthesis methods rely on statistical modeling of motion capture data. Motion capture data is a high dimensional time-series data, therefore, it is usually required to construct an expressive latent space through dimensionality reduction methods in order to reduce the computational costs of modeling such high-dimensional data and avoid the curse of dimensionality. However, different features of the motion data have intrinsically different scales and as a result we need to find a strategy to scale the features of motion data during dimensionality reduction. In this work, we propose a novel method called Scaled Functional Principal Component Analysis (SFPCA) that is able to scale the features of motion data for FPCA through a general optimization framework. Our approach can automatically adapt to different parameterizations of motion. The experimental results demonstrate that our approach performs better than standard linear and nonlinear dimensionality reduction approaches in keeping the most informative motion features according to human vision judgment.

Keywords: Human Motion Synthesis, Functional Data Analysis, Scaled Functional Principal Component Analysis

Concepts: •**Computing methodologies** → **Computer graphics;** *Animation;*

1 Introduction

Data-driven approaches for efficiently generating a wide range of natural-looking motion with different styles have been a very active research field. However, how to efficiently reuse recorded motion capture data and synthesize motion that can be easily controlled still pose challenges to data-driven motion synthesis [Min and Chai 2012; Wang and Neff 2015].

In order to reuse existing motion capture data for other tasks such as motion synthesis, it is necessary to decouple the original motion data into meaningful segments and reassemble them for new scenarios. This can be done either in a frame-wise or block-wise manner. Frame-wise methods allow the new motion to make use of all possible variations in the recorded data. Motion Graphs [Kovar et al. 2002] build graph structures to allow transition between each pair of frames, and convert the motion synthesis problem into a graph search problem. Motion Graphs work well for a small dataset containing thousands of frames. However, it does not scale well for larger datasets containing millions of frames [Kovar et al. 2002]. For large motion databases, a highly structured motion data representation is required. Min and Chai [2012] assume that although human motion appears to have infinite variations, the fundamental high-level structures are always finite. For example, normal walking can be regarded as a sequence of alternating left and right stances, and picking can be decomposed as reaching and retrieving. They construct a generative statistical graph model using structurally similar motion clips, and validate the ability of their model to interactively generate controllable, natural-looking motion on a large database.

In general, human motion is high-dimensional data because of high degree of freedom of the body. Working on such high-dimensional space is not useful in practice, so dimensionality reduction approaches are employed before statistical analysis. For motion synthesis, a good dimensionality reduction approach should be able to reconstruct the training data without noticeable error observed by a human. The latent space constructed by dimensionality reduction should have good interpolation and extrapolation quality [Quirion et al. 2008].

We construct our work based on the previous works of [Min and Chai 2012; Du et al. 2016] to apply Functional Principal Component Analysis (FPCA) on motion clips, and model the distribution of latent space with a statistical approach. For statistical analysis of motion data, it is common to represent each frame of motion as a vector of 3D root joint translation and orientation, plus orientations of other joints relative to their parents. Although this hierarchical parameterization of motion data has the advantage of preserving the actor's skeleton structure during statistical processing, it suf-

*han.du@dfki.de

MiG '16, October 10 - 12, 2016, Burlingame, CA, USA

ACM 978-1-4503-4592-7/16/10. . . $15.00
DOI: http://dx.doi.org/10.1145/2994258.2994277

fers from the fact that the translations and orientations of the joints are measured in different scales. This scaling issue may cause problems for the FPCA analysis of motion data since only a few features dominate the total variance of the motion data. Moreover, the hierarchical parameterization of motion data does not reflect the similarity of motion clips in the Euclidean joint space which is more coherent with the human's eye judgment. There are mainly two reasons for that: First, different features of motion vectors have different influence on joint positions in Euclidean space. For instance, a small change in the translation of the root joint has much more influence on the pose than a small change in the hand orientation. Second, a joint position is decided not only by the joint orientations in the kinematic chain but also by the joint offset vectors (bone between two joints). Therefore, joints with the same rotation angles may have different Euclidean positions. Consequently, it is required to design a strategy for scaling the features of motion data during FPCA process.

As the main contribution of this work, we present a novel dimensionality reduction method for motion data called Scaled FPCA to address the aforementioned challenges. Our idea is similar to previous work [Grochow et al. 2004], which introduced the scaled Gaussian Process Latent Variable Model for pose modeling. However, our problem is different because the dimensionality of a motion clip is higher compared to the dimensionality of a single pose. Additionally, the number of available samples for modeling motion clips is smaller than for modeling of poses, which could cause overfitting. We scale each feature of the motion data during FPCA to minimize the reconstruction error in the Euclidean joint space instead of feature space. Since the variances of joint positions in the Euclidean space are unique regardless of the parameterization type of the motion data, our approach can automatically adapt to different parameterizations. Moreover, our method retains the desirable variance of the motion data in Euclidean space better compared to standard FPCA, which is important for the visualization quality of the reconstructed motion.

2 Related Work

A great deal of work has been done so far on human motion modeling and synthesis in latent space. For motion synthesis purposes, an explicit inverse transform from latent space to original motion space is required. The dimensionality reduction methods for motion data can be categorized as linear and nonlinear methods.

Principal Component Analysis (PCA) [Pearson 1901] is the most widely used linear approach due to its simplicity and efficiency. Safonova et al. [2004] apply PCA on individual motion frames to optimize the motion in low dimensional space. Urtasun et al. [2004] find the latent space for a sequence of frames by using PCA on sequentially concatenated motion vectors. Forbes and Fiume [2005] propose to use a weighted PCA to reduce dimensionality of motion data. Instead of considering motion frames as a long multivariate vector, Du et al. [2016] and Min and Chai [2012] apply functional data analysis on the motion data, and construct a latent space using Functional PCA.

Most nonlinear dimensionality reduction methods, e.g. Locally Linear Embedding [Roweis and Saul 2000] and ISOMAP [Balasubramanian and Schwartz 2002], are not feasible for our work because they do not provide an explicit inverse transform to the original motion space. One exception is Gaussian Process Latent Variable Model (GPLVM) [Lawrence 2004]. GPLVM is a generalized form of PCA that can learn a nonlinear smooth mapping from the low-dimensional latent space to the observation space.

Growing research interests in GPLVM have led to many new variants of this method. In [Grochow et al. 2004], the scaled GPLVM is proposed in order to improve the original GPLVM for modeling the human poses in which the different dimensions of the observation space are of different scales. Scaled GPLVM introduces a scaling weight for each dimension of the observation space. In [Lawrence and J.Q.Candela 2006] a GPLVM with back constraints is proposed to preserve the local distances of the data from the observation space to the latent space. GPLVM keeps the dissimilar points in the observation space apart in the latent space, however, it does not guarantee that the similar points in the observation space stay close to each other in the latent space. Urtasum and Darrell [2007] introduce a discriminative prior over the latent variables in order to regard for the class labels in GPLVM. While having the generalization ability of probabilistic methods, this method can also preserve the class labels in the latent space. One of the training issues of the traditional GPLVM is that it is prone to over-fitting since no proper distribution is used for the latent variables and they are estimated as points using optimization. To address this issue, Titsias and Lawrence [2010] introduce the Bayesian GPLVM approach, which is more robust to overfitting.

3 Motion Parameterization

The goal of this work is to build a statistical motion database (Figure 1), which can take constraints derived from some high-level user interface, such as a controlled natural language [Busemann et al. 2016], to generate infinitely many new styles of natural-looking motion from finite motion capture data. Our work is based on the previous successful work in [Min and Chai 2012]. In order to be able to generate infinite new variants of finite previously-recorded motions, high level structures need to be defined for each type of motion. For example, the normal walk is defined as a combination of six kinds of atomic motion clips: leftStance, rightStance, startLeftStance, startRightStance, endLeftStance and endRightStance. These high level structures which are named motion primitives offer us an efficient and compact way to describe different behaviors.

3.1 Motion Data Preprocessing

Our motion data processing steps are similar to [Min and Chai 2012]. The long input recordings are decomposed into structurally similar clips. A good decomposition should make the dissimilarity small for clips within the same motion primitive and very large for clips in different motion primitives. The quality of motion decomposition decides the quality of our models. Motion clips within the same motion primitive could have different root positions, orientations and number of frames. We align the motion segments within the same primitive to a heuristically selected reference segment using dynamic time warping. Each frame is represented as root translation and orientation, plus joint orientations. The orientations are represented by unit quaternions. In our work, we use the approach in [Du et al. 2016] to smooth quaternions and remove singularities.

3.2 Functional Data Analysis on Motion Data

Human motion changes smoothly and continuously over time. Therefore, the motion capture data can be intrinsically represented and analyzed in the functional domain. There are two advantages to use a functional representation: First, the noise in motion capture data can be smoothed; Second, it offers a compact representation for the frame sequences to reduce redundancy in time domain due to a high frame rate. We consider motion clips as multivariate functional data and the discrete values of each dimension over frames are interpolated as a smooth function represented by a linear com-

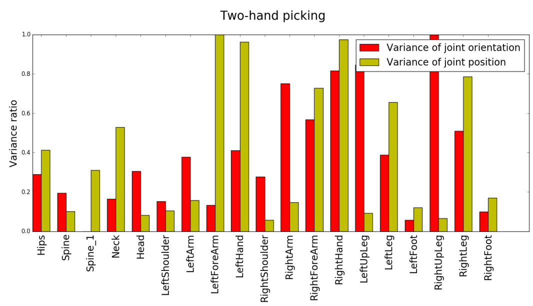

Figure 2: *Comparison of joint variations of two-hand picking.*

bination of cubic B-spline functions:

$$\{q_{i1}, ..., q_{in}\} \rightarrow q_i(t) = \sum_{k=1}^{K} c_{ik}\phi_k(t), \quad K \ll n \tag{1}$$

where, q_{ij} denotes the quaternion value of the i^{th} dimension of the motion in the j^{th} frame, $\phi_k(t)$ is the k^{th} cubic B-spline basis function and c_{ik} is the coefficient of the k^{th} B-spline basis function for the i^{th} dimension of the motion.

By applying a functional representation, each motion clip $\mathbf{y_i}$ can be represented as a vector of continuous functions instead of a very long multivariate vector.

$$\mathbf{y}_i(t) = \{\mathbf{p}_i(t), \mathbf{q}_{i1}(t), ..., \mathbf{q}_{im}(t)\}^T \tag{2}$$

where $\mathbf{p}_i(t) \in \mathbf{R}^3$ is the root translation of i^{th} motion clip $\mathbf{y_i(t)}$, and $\mathbf{q}_{ij}(t) \in \mathbf{R}^4$ is the j^{th} joint orientation represented by unit quaternions.

Since the motion data is sampled at a constant frame rate, we select equally-spaced knots for our B-spline representation. The choice of the number of B-spline basis functions K depends on the complexity of each motion type. We use the cut-off point at which increasing the number of basis functions will not significantly reduce the root mean squared error in Euclidean joint space as the number of basis functions for each motion primitive.

4 Scaled Functional Principal Component Analysis

Similar to PCA, Functional PCA reduces the dimensionality of data by finding a subspace which accounts for as much of the variability in the data as possible. For functional data, the subspace is defined by a set of orthonormal eigenfunctions $\mathbf{V}(t)$, which minimize the reconstruction error in feature space:

$$\arg\min_{\mathbf{V}(t)} \frac{1}{N} ||\mathbf{Y}(t) - \mathbf{V}(t)\mathbf{V}^T(t)\mathbf{Y}(t)||_F^2 \tag{3}$$

$$\mathbf{V}^T(t) = \begin{pmatrix} \boldsymbol{\xi}_0(t) \\ \vdots \\ \boldsymbol{\xi}_p(t) \end{pmatrix} = \begin{pmatrix} \xi_{0,\mathbf{p}}(t) & \cdots & \xi_{0,\mathbf{q}_m}(t) \\ \vdots & \ddots & \vdots \\ \xi_{p,\mathbf{p}}(t) & \cdots & \xi_{p,\mathbf{q}_m}(t) \end{pmatrix} \tag{4}$$

where $\mathbf{Y}(t)$ is the functional representation of input motion clips, $\mathbf{V}(t)$ are eigenfunctions of $\mathbf{Y}(t)$, and N is the number of motion clips.

The inner product of two vector of functions $\boldsymbol{\xi}(t)$ and $\boldsymbol{y}(t)$ is defined as:

$$\boldsymbol{\xi}(t)\boldsymbol{y}(t) = \sum_{i=1}^{m} \int \xi_i(t) y_i(t) \mathrm{d}t \tag{5}$$

However, for motion data, joints could have different variances under different parameterizations. Figure 2 compares the variance of each joint for two-hand picking measured in different spaces. The variances by using these two parameterizations are significantly different. The information reduced by FPCA because of low variance in feature space can actually cause a big difference in Euclidean joint space. Based on this observation, we propose to measure the reconstruction error between original motion and reconstructed motion in Euclidean joint space rather than feature space (6), which more corresponds to human visual observation.

$$\arg\min_{\mathbf{V}(t)} \frac{1}{N} ||f_k(\mathbf{Y}(t)) - f_k(\mathbf{V}(t)\mathbf{V}^T(t)\mathbf{Y}(t))||_F^2 \tag{6}$$

where $f_k(\mathbf{Y}(t))$ is the forward kinematic function which maps functional motion data $\mathbf{Y}(t)$ to joint positions in Euclidean space. Applying FPCA on $\mathbf{Y}(t)$ can find an optimal solution $\mathbf{V}(t)$ for (3), however, $\mathbf{V}(t)$ is not the optimal solution for (6).

Finding an analytical optimal solution for (6) is nontrivial due to the complexity of forward kinematics. In this work, we apply a simple scaling method to further reduce the reconstruction error defined by (6), similar to [Grochow et al. 2004]. We adapt their idea to our problem with two modifications. First, we apply scaling on functional data rather than discrete data, which significantly reduces the number of weights to be estimated by the factor of the number of frames and reduces the risk of overfitting. Directly scaling each dimension of the motion clip is not practical in our work. For example, for two-hand placing reach, each motion clip has 145 frames and each frame has 79 variables (Table 1). If we sequentially concatenate frames, each motion clip will have 11455 dimensions. In this case, optimizing the weight for each dimension is not only computationally expensive but overfits the data as well. Second, the weights are calculated by minimizing the average squared error in the Euclidean joint space. We use this prior knowledge to guide the weights calculation. An optimal functional space $\mathbf{V}'(t)$ is found by employing FPCA on scaled functional data $\mathbf{WY}(t)$. The weights $\mathbf{W}$ are represented as a diagonal matrix $diag(w_1, ..., w_d)$. So (6) can be rewritten as:

$$\arg\min_{\mathbf{W}} \frac{1}{N} ||f_k(\mathbf{Y}(t)) - f_k(\mathbf{W}^{-1}\mathbf{V}'(t)\mathbf{V}'^T(t)\mathbf{WY}(t))||_F^2 \tag{7}$$

The optimal weights $\mathbf{W}$ can be found by minimizing (7). We apply the numerical optimization algorithm L-BFGS-B [Byrd et al. 1995] to iteratively find the optimal weights. At each evaluation of (7) during the optimization the eigenfunctions $\mathbf{V}^k(t)$ are updated by applying FPCA on updated scaled motion data $\mathbf{W}^k\mathbf{Y}(t)$. The initial weights are set heuristically to accelerate the optimization and reduce the chance to get stuck in a local minimum. We first compare the reconstruction error of (6) using raw functional motion data and normalized functional motion data, and take the one with smaller error as an initial guess for the weights optimization.

5 Motion Modeling and Synthesis

A latent space is constructed by applying SFPCA on functional motion data $\mathbf{X} = f(\mathbf{Y}(t))$. We learn the distribution of the latent space $P(\mathbf{X})$ for each motion primitive by using Gaussian Mixture

Model (GMM). The parameters of GMM are estimated by applying Expectation-maximization (EM) algorithm. The number of Gaussians is selected by empirically evaluating a set of values to maximize Bayesian Information Criterion (BIC) score for training data in latent space $\mathbf{X}$.

Any number of motion variations with different styles can be efficiently generated by sampling the motion primitive models we construct. For constrained motion synthesis, the statistical model provides a maximum a posterior framework to support different kinds of constraints.

$$\arg\max_{\mathbf{x}_i} P(\mathbf{x}_i|\mathbf{c}) = \arg\max_{\mathbf{x}_i} P(\mathbf{c}|\mathbf{x}_i)P(\mathbf{x}_i) \tag{8}$$

where $\mathbf{x}_i$ is the target motion in latent space, $\boldsymbol{c}$ is a set of constraints, which can be joint positions, orientations, and even some high level constraints.

6 Experimental Results

In this section, we provide quantitative analysis of our approach on a fairly large motion capture database for multiple actions. We compare our method with two baseline approaches: FPCA and normalized FPCA. For FPCA, normalized FPCA and our method, we all use cubic Bspline basis. The same number of basis functions is used for each motion primitive. In addition to linear dimension reduction methods, an advanced nonlinear method Discriminative Prior Bayesian GPLVM is evaluated as well. Discriminative Prior Bayesian GPLVM combines Bayesian GPLVM and discriminative GPLVM. It is robust to overfitting, which is important for our work since the dimensionality of our data is much higher than the number of samples.

In our experiments, we implement Functional PCA described in [Ramsay and Silverman 2005]. Standard PCA is applied on the coefficients of functional data. The Python implementation of DPBayesianGPLVM in the GPy package http://github.com/SheffieldML/GPy is employed. Radial basis function (RBF) kernel is used to perform a nonlinear mapping from latent space to original space. The L-BFGS-B optimizer is chosen to optimize latent parameters. For the discriminative prior over latent space, we automatically label the training motion clips according to their pose similarity. Motion clips are parameterized as joint position splines, and clustered by the k-means++ algorithm [Arthur and Vassilvitskii 2007]. The number of clusters is set heuristically. For example, although motion clips in "walk leftStance" contain a rich variation, the main visual styles are three: walk to left, walk to right and walk forward.

The latent spaces constructed from different approaches are evaluated by two criteria. First, the low dimensional representation in latent space should be able to reconstruct original motion without observable visual difference. Therefore, the average reconstruction Euclidean error between original motion data and reconstructed motion data is compared in two spaces: feature space and Euclidean joint space. The reconstruction error in feature space can tell us how good the dimension reduction method is in reconstructing input data, reconstruction error in Euclidean joint space tells how good the dimension reduction method is in reconstructing data for visual observation. Second, the motion clips which are similar in Euclidean space should stay close in latent space as well. So we evaluate the distribution of pre-labeled samples in latent space.

Table 1: *Examples of motion primitives*

Motion primitive types	No. motion clips	No. canonical frames
walking leftStance	749	46
two-hand picking reach	378	103
two-hand carrying leftStance	280	52
two-hand transferring	82	165

6.1 Motion Capture Database

Our motion capture data is recorded by an OptiTrack system and contains 10 distinctive elementary actions performed in assembly workshop scenario, including walking (125765 frames), carrying (413090 frames), two-hand picking (410190 frames), single-hand picking (86604 frames), two-hand placing (335310 frames), single-hand placing (50159 frames), sidestep (240734 frames), screwing (165920 frames), looking around (71034 frames), two-hand transferring (18854 frames). The motion capture data is saved as BVH format. Table 1 lists details of training data for some motion primitives constructed from our motion capture database.

6.2 Reconstruction Errors

Figure 3 compares the reconstruction error of all methods for four motion primitives: walk leftStance, two-hand picking reach, two-hand carrying leftStance and two-hand transferring. DPBayesianGPLVM achieves comparably small reconstruction error for most numbers of dimensions. However, there are exceptions where the reconstruction errors go up with increasing numbers of dimensions, and the error is relatively large compared to other dimensions. For FPCA and normalized FPCA, the reconstruction error monotonically decreases with more dimensions. FPCA yields lower reconstruction errors than SFPCA and normalized FPCA in feature space. Our SFPCA method gets the lowest reconstruction error in Euclidean joint space for most dimensions. Normalized FPCA has the largest reconstruction error for most dimensions, however, it is noticeable that the reconstruction error for two-hand picking in Euclidean joint space is smaller than FPCA. Two-hand transferring has a much smaller reconstruction error in feature space for all methods than other three motion primitives. This is because transferring has upper body movements without root translation. The joint orientations represented by quaternions take a small portion of the variance in feature space. However, the loss in Euclidean joint space is considerable compared to the small loss in feature space.

From our experiments, we noticed that using FPCA and DPBayesianGPLVM 10 dimensions are sufficient to achieve a reconstruction error of less than 0.1 in feature space for most motion primitives. However, the reconstruction error in Euclidean joint space varies significantly. This observation indicates that deciding the number of latent variables in feature space might be misleading for some motion primitives.

6.3 Latent Space Analysis

Figure 4 shows the distribution of pre-labeled motion clips in the latent space for four motion primitives. The original motion data is reduced to 20 dimensions using four testing methods separately. Here, we choose 20 dimensions to make the reconstruction errors fairly small in both feature space and Euclidean joint space for all motion primitives. The samples in latent space are visualized in 2D space using the t-SNE algorithm [van der Maaten and Hinton 2008], which is powerful for keeping the local structure of the data.

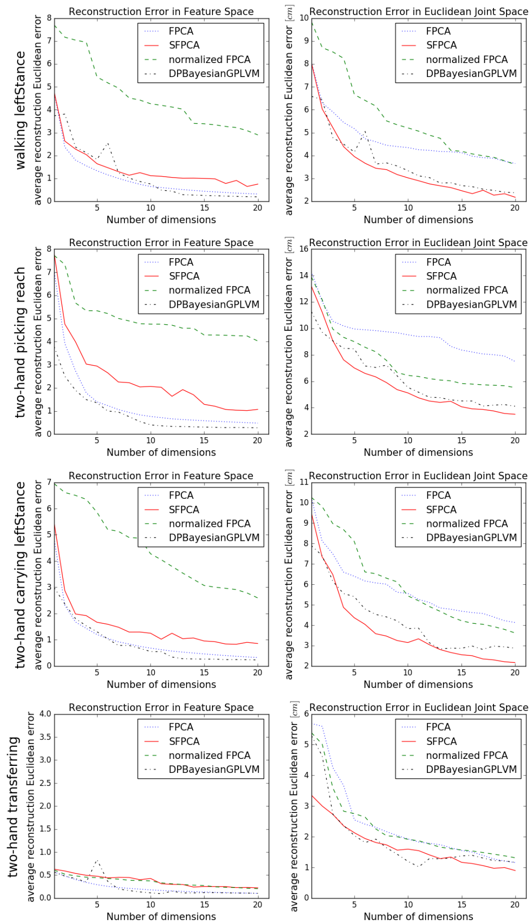

Figure 3: *Comparison of reconstruction error of four dimension reduction methods: Each curve shows the change of average reconstruction Euclidean error with the increasing of number of dimensions.*

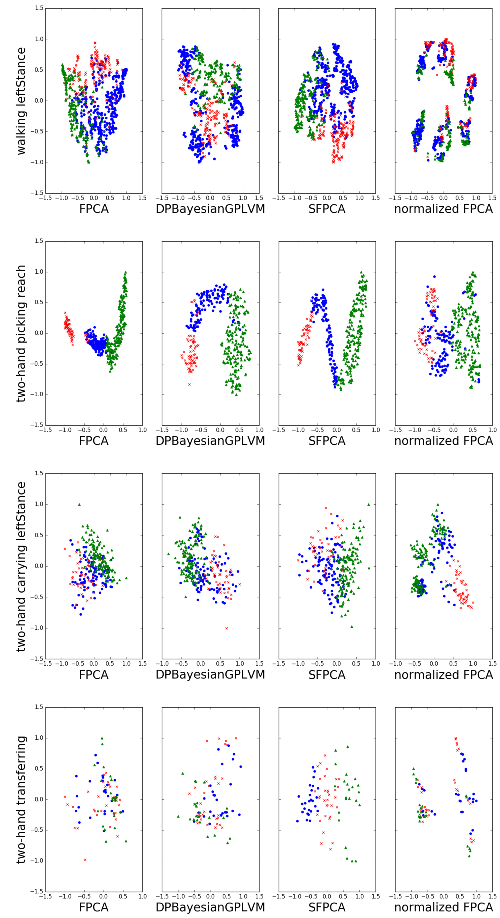

Figure 4: *Clusters of latent space constructed from four dimension reduction methods. For walk leftStance and two-hand carrying leftStance, the red cross is moving forward, the blue circle is moving to the left and the green triangle is moving to the right. For two-hand picking reach, the red cross is picking from bottom, the blue circle is picking forward and the green triangle is picking from top. For two-hand transferring, the blue circle is transferring at the right side, the red cross is transferring in front and green triangle is transferring at the left side of the body.*

7 Discussion

Overall, FPCA can find an optimal linear mapping to minimize the average squared Euclidean error in functional space. DPBayesianGPLVM, for most cases, outperforms FPCA since it takes the FPCA result as initial guess, and optimizes latent variables to maximize the posterior $p(\mathbf{Y}|\mathbf{X})$. However, the optimization method does not guarantee to find a global optimum so that it may converge early in a local optimum. Both methods achieve the optimal results in the feature space. However, their performance for motion dimensionality reduction depend on the choice of parameterization. For motion synthesis, we are interested in the reconstruction error in Euclidean joint space, which more corresponds to visual observation. Normalized FPCA provides a naive scaling without taking prior knowledge into account, however, for some motion primitives like both-hand picking, it outperforms FPCA, which suggests that scaling features makes sense for motion data. Our SFPCA takes the nonlinear mapping from feature space to Euclidean joint space as prior knowledge for weights optimization. Therefore, it can achieve the lowest reconstruction error in Euclidean joint space for most cases. Since L-BFGS-B is a local optimizer, we also have the risk to converge on local optimum. That's why the reconstruction errors are not monotonically decreasing in figure 3.

For locomotion motion primitives, such as walking and carrying, the dominate variance in both feature and Euclidean space is the root translation while the variance of the joint orientations is small. Since FPCA put most efforts to keep variance in given space, FPCA results are still better than normalized PCA. For the local structure in latent space, FPCA also has a similar result as SFPCA. Motion clips which are close in Euclidean joint space are still close in latent space. However, DPBayesianGPLVM and Normalized FPCA fail to keep that. For two-hand carrying, no method can clearly keep local structure in latent space. This could indicate that our heuristic number selection of clusters based on walking direction is not sufficient to cover the major variance in two-hand carrying since the variance of the upper body should not be ignored. FPCA and normalized FPCA have a similar reconstruction error for two-hand carrying in Euclidean space which shows that the pose variance is larger than for walking.

For manipulation motion primitives, for example, two-hand picking and transferring, the pose variation is larger than the root translation. SFPCA outperforms other approaches in both Euclidean reconstruction error and local structure preserving in latent space.

Two-hand transferring is a motion primitive which has no change of the root translation. SFPCA clearly separates the clusters in latent space, which shows that our approach has good performance to reduce the dimension of motion data according to the variance in Euclidean joint space.

In addition, the statistical motion database based on latent space constructed from SFPCA provides a compact representation of the original motion capture data. Any number of new motion can be generated from our motion primitive models, and the variation of generated motion is larger than training data. The motion capture data used for our work has a size of 4.2 GB, however, our motion primitive models only require 41 MB, which is 100 times smaller. Our motion synthesis result can be viewed on the INTERACT project homepage http://www.interact-fp7.eu.

8 Conclusion

In this work, we analyze different motion data parameterization approaches, and their corresponding visual effects observed from Euclidean joint space. A latent space motion modeling approach is presented based on Scaled Functional Principal Component Analysis. We conclude that functional data analysis can represent motion clips in a more generative and compact way by reducing the redundancy of the data in time domain. Scaling motion data based on visual similarity in Euclidean joint space can guide the mapping of motion data from high dimensional space to latent space in a more meaningful sense. We demonstrate that taking this prior knowledge into dimension reduction results in an improved latent space regarding reconstruction error in Euclidean space and local structure preservation compared to no and naive scaling.

We show the power of our approach on modeling motion clips which consist of root translations and joint orientations represented by quaternions. In principle, our approach is general for different motion parameterizations. We construct our approach for linear dimensionality reduction on functional motion data. However, we believe that nonlinear dimensionality reduction approaches for motion data can also benefit from a functional data representation and feature scaling based on similarity in Euclidean joint space.

Acknowledgements

We would like to thank the reviewers for their valuable comments. This work is funded by the European Union FP7 project INTERACT under grant agreement No 611007 and the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 642841.

References

ARTHUR, D., AND VASSILVITSKII, S. 2007. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, SODA '07, ACM, 1027–1035.

BALASUBRAMANIAN, M., AND SCHWARTZ, E. 2002. The isomap algorithm and topological stability. *Science 295(5552)*, 7.

BUSEMANN, S., STEFFEN, J., AND HERRMANN, E. 2016. Interactive planning of manual assembly operations: From language to motion. *In Procedia CIRP 41*, 224–229.

BYRD, R. H., LU, P., NOCEDAL, J., AND ZHU, C. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing 16*, 5, 1190–1208.

DU, H., MANNS, M., HERRMANN, E., AND FISCHER, K. 2016. Joint angle data representation for data driven human motion synthesis. *In Procedia CIRP 41*, 746–751.

FORBES, K., AND FIUME, E. 2005. An efficient search algorithm for motion data using weighted pca. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, 67–76.

GROCHOW, K., MARTIN, S., HERTZMANN, A., AND POPOVIC, Z. 2004. Style-based inverse kinematics. *In ACM Transactions on Graphics (Proc. of SIGGRAPH 2004)*, 522–531.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. In *ACM transactions on graphics (TOG)*, vol. 21, ACM, 473–482.

LAWRENCE, N., AND J.Q.CANDELA. 2006. Local distance preservation in the gp-lvm through back constraints. *Omnipress 23*.

LAWRENCE, N. 2004. Gaussian process latent variable models for visualisation of high dimensional data. *Proc. of NIPS 2003*.

MIN, J., AND CHAI, J. 2012. Motion graphs++: A compact generative model for semantic motion analysis and synthesis. *In ACM Transactions on Graphics 31*, 6, 153:1–153:12.

PEARSON, K. 1901. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine 2*, 6, 559–572.

QUIRION, S., DUCHESNE, C., LAURENDEAU, D., AND MARCHAND, M. 2008. Comparing gplvm approaches for dimensionality reduction in character animation.

RAMSAY, J. O., AND SILVERMAN, B. W. 2005. *Functional Data Analysis*. Springer.

ROWEIS, S., AND SAUL, L. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500, 2319–2323.

SAFONOVA, A., HODGINS, J. K., AND POLLARD, N. S. 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics (TOG) 23*, 3, 514–521.

TITSIAS, M., AND LAWRENCE, N. 2010. Bayesian gaussian process latent variable model. *Proceedings of the Thirteenth International Workshop on Artificial Intelligence and Statistics 9*, 13–16.

URTASUM, R., AND DARRELL, T. 2007. Discriminative gaussian process latent variable model for classification. *In ICML*, 927–934.

URTASUN, R., GLARDON, P., BOULIC, R., THALMANN, D., AND FUA, P. 2004. Style-based motion synthesis. *Computer Graphics Forum 23*, 4, 799–812.

VAN DER MAATEN, L., AND HINTON, G. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research 9(Nov)*, 2431–2456.

WANG, Y., AND NEFF, M. 2015. Deep signatures for indexing and retrieval in large motion databases. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, ACM, 37–45.