Comparing GPLVM Approaches for Dimensionality Reduction in Character Animation

Sébastien Quirion[1]   Chantale Duchesne[1]   Denis Laurendeau[1]   Mario Marchand[2]

[1] Computer Vision and Systems Laboratory
Dept. of Electrical and Computer Engineering
Laval Univ., Ste-Foy, Qc, Canada, G1V 0A6

[2] Laboratory of Computational Intelligence
Dept. of Comp. Science and Software Engineering
Laval Univ., Ste-Foy, Qc, Canada, G1V 0A6

squirion@gel.ulaval.ca, chantale.duchesne.1@ulaval.ca, laurend@gel.ulaval.ca,
mario.marchand@ift.ulaval.ca

ABSTRACT

Gaussian Process Latent Variable Models (GPLVMs) have been found to allow dramatic dimensionality reduction in character animations, often yielding two-dimensional or three-dimensional spaces from which the animation can be retrieved without perceptible alterations. Recently, many researchers have used this approach and improved on it for their purposes, thus creating a number of GPLVM-based approaches. The current paper introduces the main concepts behind GPLVMs and introduces its most widely known variants. Each approach is then compared based on various criteria pertaining to the task of dimensionality reduction in character animation. In the light of our experiments, no single approach is preferred over all others in all respects. Depending whether dimensionality reduction is used for compression purposes, to interpolate new natural looking poses or to synthesize entirely new motions, different approaches will be preferred.

Keywords
Character Animation, Gaussian Process, GPLVM, SGPLVM, GPDM, B-GPDM, Dimensionality Reduction, Degrees of Freedom, Nonlinear Transform, PCA.

1. INTRODUCTION

Animating realistic human-like characters usually requires handling a large number of degrees of freedom (DOF). The skeleton we use in our own research is a good example, with 44 active DOFs (see Figure 1). Considering each of the d DOFs as a dimension, each possible pose of the character may be seen as a vector in a d-dimensional space.

Such high-dimensional spaces are generally not useful in practice since they usually translate into an increased amount of work, both for the animator and for the computer, and require more storage space. Moreover, they do not provide any insight regarding what poses look natural or regarding how one should interpolate between two known poses (i.e. linear interpolations in these spaces rarely look natural).
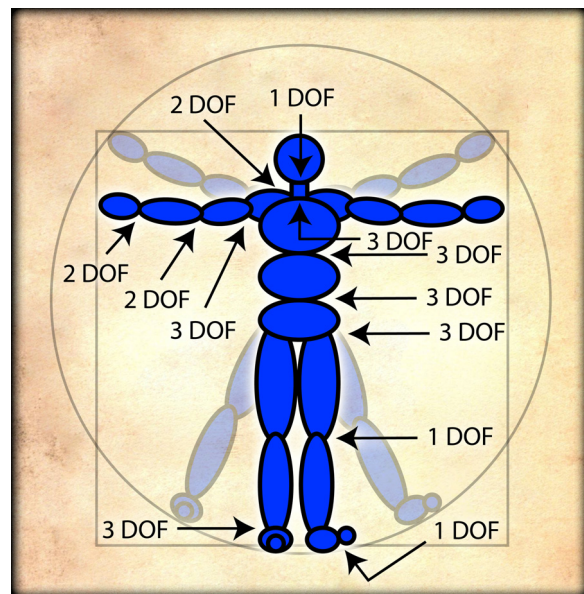
Figure 1. The DOFs for the human model used in our experiments

For these reasons, among others, animators may well be interested in more compact and more informative representation spaces. By learning motion-specific models, one might use the correlation between values

at each DOF to reduce the number of effective DOFs while preserving the resulting animation almost intact. As will be shown later, GPLVM-based techniques are especially well suited for this task.

In this paper, we present the GPLVM approach and its most widely known variants. We then compare their performances on keyframe animations extracted from motion capture experiments. For comparison purposes, we perform the same experiments with Principal Components Analysis (PCA). This paper is in line with our research work, which consists in animating virtual characters using forward dynamics under a physics simulation. To this end, GPLVMs are used to learn kinematic motion models and to interpolate new natural looking poses.

2. RELATED WORK

Principal Components Analysis (PCA), which is arguably the best known dimensionality reduction technique, finds its origins early in the 20th century. Pearson [Pea01a] and Hotelling [Hot33a] are generally credited as providing the earliest descriptions of PCA. (Source: [Jol02a])

The concept of using Gaussian processes for the purpose of dimensionality reduction, through Gaussian Process Latent Variable Models (GPLVM), has been introduced by Lawrence in 2003 [Law03a] and 2005 [Law05a]. The main idea behind the GPLVMs is to find a non-linear function that smoothly maps low-dimensional latent-space vectors to a high-dimensional observation-space. In 2006, Lawrence et al. improved on the GPLVM by proposing back-constraints which enforce the conservation of local distances from the observation-space over to the latent-space [Law06a].

In 2004, Grochow et al. proposed the Scaled Gaussian Process Latent Variable Model (SGPLVM) [Gro04a]. This approach improves on the original GPLVM by learning a scaling factor for each observation-space dimension, thus expressing the process of learning a model in a normalized observation-space. Their work was also instrumental in highlighting the potential of GPLVM approaches in the field of realistic character animation.

In 2005, Wang et al. presented the Gaussian Process Dynamical Model (GPDM) [Wan05a]. While GPLVMs may be used on any kind of multi-dimensional data, GPDMs are specially designed to handle datasets exhibiting chronological relations between successive data points, such as keyframe animations. As with the SGPLVM, the GPDM approach uses scaling factors for each of the observation-space dimension.

In 2006, Urtasun et al. proposed the Balanced GPDM (B-GPDM) [Urt06a]. This contribution is essentially a slight alteration to the objective function used during the learning of the model. This revised objective function amplifies the importance of smoothness of the latent-space in the final model. It was studied in more details by Wang et al. [Wan07a].

3. PROBLEM DEFINITION

Consider a keyframe animation of m frames for a character with d DOFs. This animation may be seen as a d-by-m matrix Y, where each of the m columns represents a pose of the character and each of the d rows represents the trajectory taken by a given DOF over the course of the animation. The observation-space for the poses of the character is d-dimensional. We wish to derive a transformation f such that

$$X = f(Y),\qquad(1)$$

where X is the q-by-m matrix representing the m frames of the animation in the latent space, with q < d. Moreover, to be of any practical use, one must be able to reverse this transformation, at least approximately, through a transformation g defined by

$$Y' = g(X),\qquad(2)$$

where Y' is the reconstructed d-by-m animation matrix. The transformation g is required because poses in the latent space may not be applied directly to the character. It is important to note that, while the entire animation is generally needed in order to produce satisfying transformations f and g, these transformations can then be applied to single poses in the observation/latent-space and not necessarily on the whole animation.

Evaluation Criteria

Now, the problem definition is very large and allows for many different solutions, most of which would not produce interesting results. Thus, specific and measurable criteria need to be defined that will allow the evaluation of the quality of a given solution.

The criteria we deem to be the most important are :
- **Dimensionality reduction potential:**
  For values of q as small as possible, the distance between an observed pose y and its reconstruction $y'=g(f(y))$ must be small enough not to be noticeable by a human observer;

- **Learning/synthesis computing time:**
  The transformation g should be as fast as possible since single poses should ideally be synthesized many times per second in order to allow for interactive applications. Another factor is the learning time for the transformations f and g;

- **Interpolation quality:**
  A linear interpolation between two known poses in the latent space should translate to a natural-looking intermediate pose in the observation space;

- **Generalization potential:**
  As animations with a high-density of keyframes are not always available, solutions generalizing quality transformations from few data points are of great interest;

- **Ease of visualization:**
  The latent space should allow quick and easy visualization of an entire motion. In order to be shown on static media, the latent space should be 2-dimensional or 3-dimensional at most. A 1-dimensional space is not interesting as it is not suited to represent generic cyclic motions as closed paths in the latent space;

- **Extrapolation quality:**
  It should be possible to generate genuinely new motions from a model by sampling the latent space near known poses. Such extrapolated motions should look fluid and natural.

4. APPROACHES

This section briefly introduces the reader to each of the compared approaches. However, the reader is referred to the original papers describing each approach for a more thorough explanation.

Principal Components Analysis

Using mean-removed data Y, the PCA consists in finding the rotation matrix $R_{d,d}$ that aligns the x_1-axis of the d-dimensional space with the direction of greatest variance in the data, the x_2-axis with the direction of second greatest variance and so on. Figure 2 illustrates this process on fictive 2-dimensional data. This matrix R can be found analytically by finding the eigenvectors of Y.

Variance along an axis gives a measure of the amount of information provided by that axis. In order to reduce the space dimensionality while maintaining enough information for accurate reconstruction, the axes associated with the least variances are removed from the transformed space. Only the q first rows of R are kept and to give a transformation matrix $T_{q,d}$. The transformation from d-dimensional observation space to the q-dimensional latent space is given by

$$x = Ty, \qquad (3)$$

and the transformation back to observation space is given by the transpose of the transformation matrix
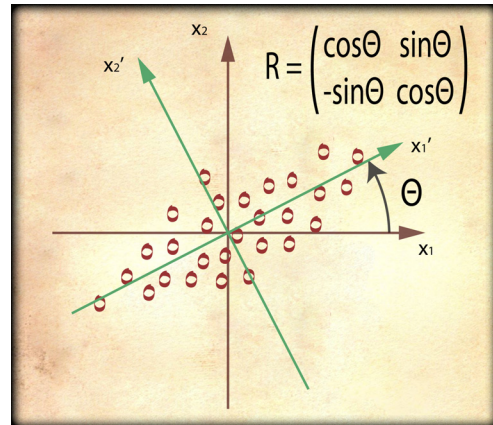
$$y' = T^t x. \qquad (4)$$

Figure 2. An illustration of PCA on 2D data

GPLVM-based Approaches

Gaussian processes are the function equivalent of Gaussian random variables. Both obey to a given probability density function, but a Gaussian random variable describes a single variable while a Gaussian process describes a whole function, which can be seen as an infinite number of variables. Gaussian random variables are defined by their mean value and their variance. In the same fashion, Gaussian processes (GPs) are defined by a mean function $\mu : \Re^q \to \Re^d$ and a covariance function $k : \Re^q \to \Re^d$, thus describing a distribution over all functions $g : \Re^q \to \Re^d$.

Gaussian processes can be used for the regression of a function on known data points in an arbitrary space. However, while *traditional* regression adjusts a single function on known data points, GP regression adjusts a distribution over a space of functions with respect to those data points. This is done by adjusting the parameters of the covariance function in order to maximize the likelihood of the observed data given the GP. This optimization favors the simplest GPs among all those explaining the observed data and thus follows Occam's razor principle.[1]

GPLVM-based approaches aim at constructing a q-dimensional latent space for d-dimensional data by learning a Gaussian process on a training set of data points. The functions considered by the GPLVM represent the transformation from latent space to observation space, which are candidates for the g function from equation (2).

GPLVMs represent the original implementation of this idea while other GPLVM-based approaches

---

[1] For a complete introduction to Gaussian Processes, the reader is referred to the book *Gaussian Processes for Machine Learning* [Ras06a], which is freely available in electronic format at: www.gaussianprocess.org/gpml/

propose slight improvements. Contrary to PCA, all of these approaches allow non-linear transformations of the data. Removing the linearity constraint usually allows far better results.

In the literature, the covariance function k is often chosen to be a Radial Basis Function (RBF) kernel for which the parameters have to be adjusted. However, any other positive definite Mercer kernel could be used. We will consider k to be a generic positive definite Mercer kernel.

4.1.1 GPLVM

In order to perform GP regression (i.e. adjusting the parameters of the kernel), we theoretically need to know both the observed data Y and the latent space data X. As X is not known *a priori*, an initial estimation is given using PCA (see equation (3)). Once X has been initialized, GP regressions and corrected estimations of X are performed iteratively until convergence has been achieved or until a maximum number of iterations has been reached.

To perform the GP regression, the likelihood of the GP and X given Y has to be maximized with respect to the parameters of the kernel.[2] This likelihood function is chosen so as to favor smooth mappings from latent space to observation space. To reassess the values of the vectors x_i, they are chosen to maximize the likelihood of X given the GP and Y. GPLVMs were first introduced in [Law03a].

For large datasets (i.e. large values of m), one may reduce the computational complexity of these optimizations by performing the GP regression using only an *active* subset of X and Y, reassessing only the inactive subset of X and choosing a different active subset for the next iteration. With this approach, each x_i may be optimized independently.

4.1.2 GPLVM with back-constraints

As presented so far, GPLVMs enforce the preservation of local proximities from latent to observation space. In other words, they insure that close poses in X are kept close in Y. This implies that GPLVMs preserve dissimilarities from observation to latent space. In other words, far apart poses in Y will be kept far apart in X. However, nothing so far prevents two similar poses in observation space to become two distant points in latent space, thus creating discontinuities in the latent space.

In order to enforce the conservation of local proximities from observation to latent space, latent space data x_i can be replaced by a function f that maps observation space data to latent space data: $x_i =$

$f(y_i)$. By optimizing over the parameters of f instead of directly optimizing the vectors x_i, a smoother mapping from Y to X is obtained. In [Law06a], Lawrence et al. present two such mappings. The first is a multilayer perceptron, later referred to as the MLP back-constraint, and the second is a RBF kernel based mapping, later referred to as the KBR back-constraint.

4.1.3 SGPLVM

Scaled GPLVMs introduce an important, yet simple, improvement over standard GPLVMs. This improvement consists in evaluating the effect of varying y along each observation space axis and scaling the values of Y accordingly, to obtain an even distribution of poses along all axes of the latent space. The different scales are initialized at unit value and are adjusted as parameters of the GP through a modified likelihood function. The likelihood of X given the GP and Y is also modified to consider these scales. SGPLVMs were introduced in [Gro04a].

4.1.4 GPDM

Proposed in [Wan05a], Gaussian Process Dynamical Models are specially designed to handle dynamical processes, such as keyframe animations. They improve on SGPLVMs by the addition of a dynamical model to the latent variable model. This dynamical model also takes the form of a Gaussian process but, instead of providing a mapping from latent to observation space, it provides a mapping from latent points happening at time $t-1$ to latent points happening at time t.

If the original mapping from observation to latent space, obtained from PCA, preserves local proximities in most cases, the dynamical model will enforce the conservation of local proximities. Loosely speaking, if two consecutive poses are nearby in X, the dynamical model will favor a nearby position for the following pose.

Depending on the number of previous values considered, the dynamical model may be defined to model the similarity between consecutive poses, the velocity, the acceleration or higher-order dynamical relations. In our experiments, it only considers the previous pose and therefore models the similarity between poses.

The likelihood function of the GPDM is similar to that of the SGPLVM multiplied by the likelihood of the dynamical model.

4.1.5 B-GPDM

Since q is generally much smaller than d, the objective function of the GPDM, its negative log-likelihood, gives less importance to the terms coming

from the dynamical model. Balanced GPDMs further improve on the GPDM simply by adding a scaling factor to these terms. This translates into a smoother mapping from observation to latent space, as the preservation of local proximities is further enforced. B-GPDMs were proposed in [Urt06a].

5. PERFORMANCES

This section presents the various experiments we performed on each technique. Lawrence's Matlab code was used for the implementation of the GPLVM, including the MLP and KBR back constraints, and for the SGPLVM. Wang's Matlab code was used for the GPDM and the B-GPDM.

The experiments were performed on two motion capture animations. The first one, *Walk*, is an animation of many walk cycles with 63 actuated DOFs and 581 frames (or poses) at 120fps. The second one, *Basketball*, is an animation of a few dribbles followed by a free throw and cheering. It animates 51 actuated DOFs and has 209 frames at 30fps. While using a greater number of test sets could have been interesting, from our experience the observations made on these animations can be generalized over to most natural human motions, cyclic or not. Moreover, using only two test animations allows for a more succinct presentation.

In our experiments, the methods were tested as presented in this paper, not using method-specific heuristics. For instance, in [Gro04a] the SGPLVM uses feature vectors as y_i, which include velocity and acceleration informations. Our experiments used only DOF values as the vectors y_i. Also, we do not use the *active set* heuristic proposed in [Law03a]. Instead, we optimize the GP parameters with respect to all points in X and then optimize all points in X with respect to the GP. Thus, each technique is not evaluated at its best, but all techniques are compared on equal grounds.

In all cases, the values of X were initialized by PCA and all GP models used RBF kernels, including the dynamical model of the GPDM and B-GPDM. The optimization of the negative log-likelihood functions was performed by the scaled conjugate gradient algorithm, as implemented in NETLAB, with a maximum of 200 iterations.

The error of a model is given by the mean over all poses of the absolute difference between the angles from observed pose y_i and reconstructed pose y_i'. The error measure is given in degrees. One may notice that this error measure does not account for the different relative effects of the various DOFs on the overall pose. While this is a drawback, very low error values do correspond to seamless reconstructions to a human observer.

Dimensionality Reduction Potential / Ease of Visualization

To evaluate the dimensionality reduction potential of each approach, we varied q, the dimension of the latent space, and evaluated the error of the models obtained using this value for q.

We empirically set a threshold $\tau = 0.5°$ under which the reconstructed motion is virtually undistinguishable from the original motion for a human observer.[3] A model with an error below this threshold is therefore considered to be a valid model. Furthermore, if the model achieves an error rate below τ for $q=3$ or $q=2$, the model is said to respect the ease of visualization criterion.

Figure 3 and 4 give the error rates obtained from each method for increasing values of q on the *Walk* sequence and the *Basketball* sequence respectively. Due both to its acyclic nature and to its lower sampling frequency (30fps), the *Basketball* sequence provides a greater challenge, which translates into greater mean errors to most techniques.
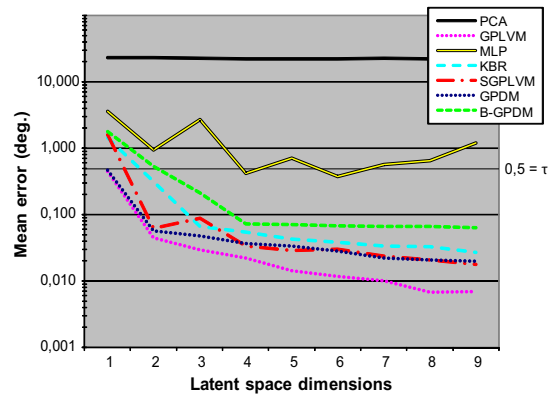

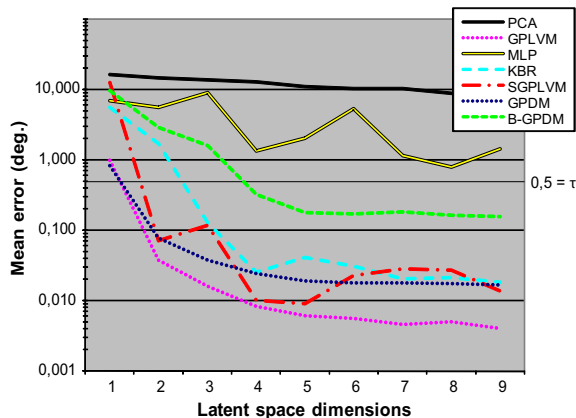**Figure 3. Mean error against number of latent dimensions for the Walk sequence**

During our experiments, we observed that the MLP and the SGPLVM approaches were more prone to converge early to a local minimum corresponding to a model of poor quality. This explains why their mean error does not necessarily decrease as q increases. Moreover, as will be illustrated later, MLP naturally learns latent spaces where the poses are constrained along a handful of line segments. This explains why it generally obtains greater error rates than the other GPLVM-based approaches.

In both cases, PCA is far above the threshold even for $q=9$. In fact, it did not reach the threshold until

---

[3] As different DOFs affect the motion on different scales, two models with similar error values might provide different qualities of reconstruction. However, such a low error threshold insures a satisfactory reconstruction for all but the most erratic models.

values of q greater than 40. MLP aside, all GPLVM-based techniques met the threshold τ on the *Walk* sequence with only 2-dimensional or 3-dimensional latent spaces. On the *Basketball* sequence however, the B-GPDM reached the threshold only starting at a 4-dimensional latent space. In both sequences, B-GPDM clearly has slightly higher mean errors. This is the effect of the emphasis that is put on learning a smooth latent space, at the expense of a more precise fitting of the training poses.

Figure 4. Mean error against number of latent dimensions for the Basketball sequence
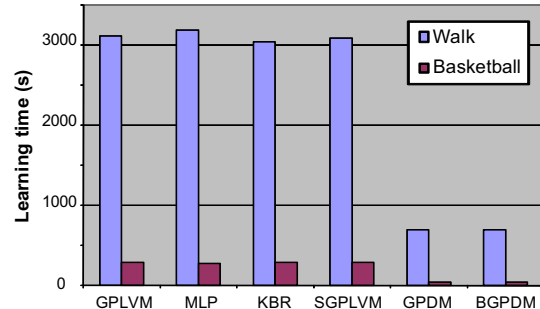
The GPLVM is also interesting. In both cases it performs better than most techniques that are, in fact, improvements over the GPLVM. This is due to the nature of this test, which does not penalize overfitting. Since the error is computed using the same poses that were used during the learning phase, overfitting these training poses may provide lower error rates. On the other hand, techniques that enforce a smoother latent space, such as B-GPDM, avoid overfitting and may obtain somewhat higher error rates. That said, if dimensionality reduction is performed only with the intention of reconstructing exactly the learned poses, GPLVM should be preferred over its alternatives.

Learning / Synthesis Computing Time

Using $q=3$, we evaluated the computing time required to learn the model (Figure 5) and to synthesize a pose given a point x in the latent space. These times were obtained on an Intel E6600 Core 2 Duo processor running at 2.4GHz. The synthesis times are averages over 200 trials.

As shown by Figure 5, the learning times for GPDM and B-GPDM appear to be significantly faster than that of the GPLVM, however this is simply due to more optimized Matlab code and not due to lower computational complexity. In fact, Lawrence's implementation of the GPDM has roughly the same learning time as the GPLVM. Considering this, all

GPLVM-based techniques require loosely the same amount of time to learn the model. This is expected as these techniques mainly differ from each other by the function being optimized during learning and not by the optimization mechanics. While these objective functions play a significant part in the total computing time, their evaluation takes roughly the same amount of time. Note that PCA is not shown in Figure 5, as its times were well below one second.

Figure 5. Learning times on both sequences

With the exception of PCA, which is significantly faster, pose synthesis times are also similar across all techniques. For GPLVM-based approaches, pose synthesis times varied between 19.0ms and 19.2ms for the *Walk* sequence and between 2.3ms and 2.5ms for the *Basketball* sequence. Once again, this is expected as the pose synthesis process is almost identical for all techniques. Using PCA, pose synthesis took less than 0.1ms on both sequences.
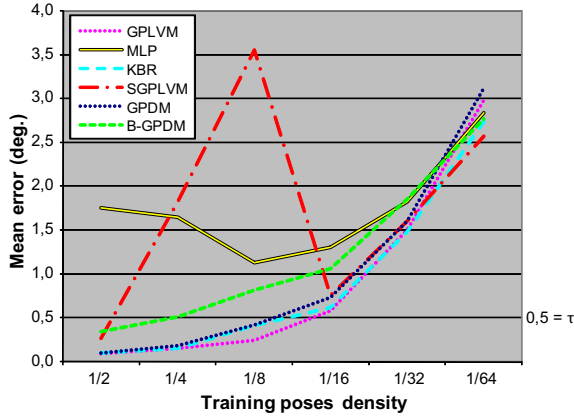
These results clearly show that the main factor in the computing time, both for model learning and for pose synthesis, is the number of poses on which the model is learned. At 209 frames, the *Basketball* sequence allows significantly faster times than the *Walk* sequence at 581 frames. In time critical applications, this should be motivation enough to use the *active set* heuristic proposed in [Law03a] or to use only a subset of the frames when learning the model.

Interpolation Quality / Generalization Potential

For this portion of the experimentation, the frames from the motion capture animations were evenly down-sampled. The models were learned using this subset of the frames and then evaluated exclusively on the frames that were left aside, again with $q=3$. This allows the interpolation of poses for which we have a ground truth. Interpolation quality and extrapolation quality were not tested on PCA as its models for $q=3$ had too high error rates when we tested for dimensionality reduction potential.

To assess the generalization quality of the models, all approaches were tested with different levels of down-sampling: using only every 2nd frame; every 4th

frame; every 8th frame; and so on. Better interpolation from fewer data points indicates better generalization potential. Figure 6 presents the results of these experimentations on the *Walk* sequence. The results for the *Basketball* sequence are not shown, but were similar in tendencies with much higher error rates.

Figure 6. Mean error against decreasing training set size for *q* = 3 on the Walk sequence

As Figure 6 shows, the GPLVM, KBR and GPDM all provide very good generalization and follow each other closely as the training set of poses gets smaller. With a training set composed of only 1/8th of the total frames, these three methods still remain under the threshold and provide seamless interpolations. As noted earlier, the SGPLVM and the MLP are prone to early convergence to a local minimum and this plagues these approaches in the quality of their interpolations as well. Finally, the B-GPDM follows the curve of the best approaches, but with a slightly higher mean error. Once again, this is due to the emphasis that is put on latent space smoothness during the learning phase.

Extrapolation Quality

We have not yet devised an objective measure for extrapolation quality. However, unlike interpolation, extrapolation depends directly on the conservation of distances by the latent space. That is, two nearby poses in observation space should be nearby in latent space and, likewise, two distant poses in observation space should be distant in latent space. Otherwise, moving a point in latent space with constant speed could represent a motion of varying speeds in observation space. Interpolation does not require these criteria as the new pose is expressed in terms of relative distance between known latent points.

In order to evaluate the extrapolation quality of a given model, one may observe the sequence of poses in the latent space. Indeed, as the motions were sampled at a constant frequency, the faster portions

of the motion should be represented by more distant points while the slower portions should be represented by closer points. Figure 7 presents the poses of the *Walk* sequence in 2-dimensional latent spaces obtained with each GPLVM-based approach.

As shown by this figure, the B-GPLVM generally presents the smoothest path of all, closely followed by KBR, which exhibits only a handful of discontinuities in the sequence of poses. In both models, the slowest portions of the motion (i.e. the end of each leg swing) can be identified by the points that are closer together and the biggest changes in direction (i.e. when arms and legs decelerate and then start moving in the opposite directions) is clearly represented by the strongest curves in the path of the sequence. Finally, in both models, the cyclic nature of the motion is obvious. From this subjective analysis, one could expect to extrapolate natural looking motions from these models.
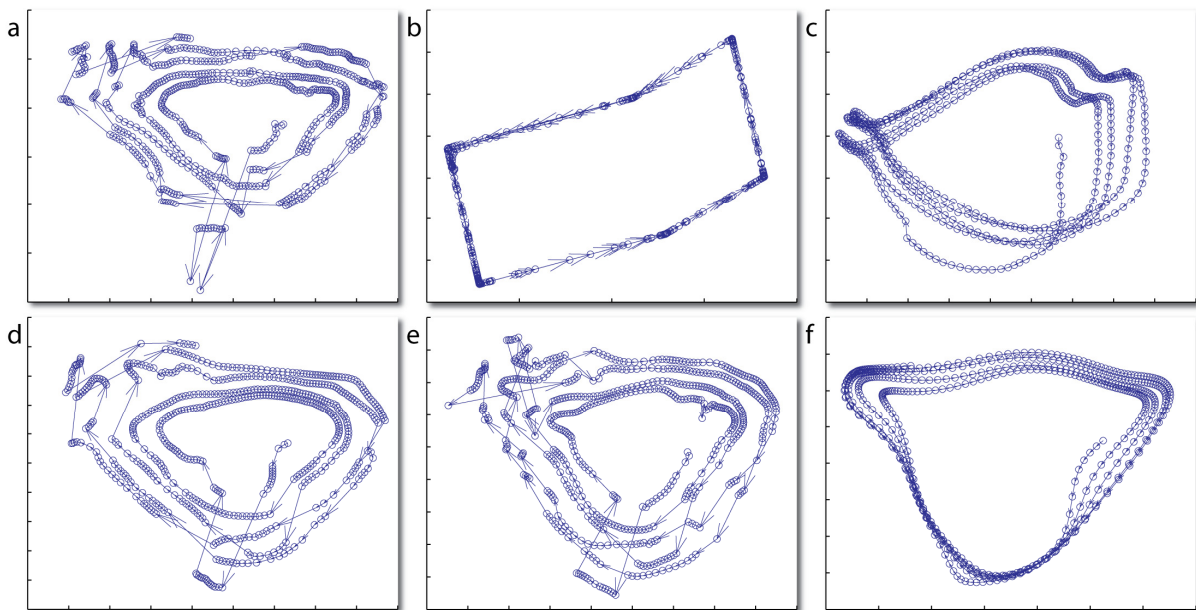
In our experiments, GPLVM, SGPLVM and GPDM all obtain latent spaces of similar smoothness. With a fair share of discontinuities, they still exhibit the cyclic nature of the motion and provide a few hints about its different phases. Finally, MLP naturally converges to paths formed of connected line segments, which hardly convey the organic nature of the motion. Moreover, as the arrows of different length show, these paths are also plagued with jerkiness.

6. CONCLUSIONS

While all GPLVM-based approaches perform better than PCA, no approach is clearly better than all others in all respects. Depending on the task at hand, different choices should be made. For instance, if dimensionality reduction is used for compression purposes, GPLVM and GPDM will allow accurate reconstruction of the original poses as well as interpolation of natural looking poses. On the other hand, if a smoother latent space is necessary, to synthesize entirely new motions as in [Urt06a] for example, the B-GPDM should be preferred.

From our experiments, the GPLVM with KBR back-constraint seems to offers the best balance. It provides better fitting to the training poses than the B-GPDM and a smoother latent space than the GPLVM and GPDM.

The authors insist on the fact that the approaches were tested on equal grounds, not using any method-specific heuristic. The results should therefore be seen as relative performances. It should also be noted that these heuristics, like the feature vector proposed in [Gro04a] and the *active set* proposed in [Law03a], could be adapted to any GPLVM-based approach and possibly improve their performance.

Figure 7. 2-dimensional latent spaces obtained on the *Walk* sequence with: GPLVM (a), MLP (b), KBR (c), SGPLVM (d), GPDM (e) and B-GPDM (f)

Finally, it may be argued that some pose synthesis times are too slow to allow interactive applications. This could easily be corrected by using an optimized C/C++ implementation of the approach and by using the *active set* heuristic.

8. REFERENCES

[Gro04a] Grochow, K., Martin, S.L., Hertzmann, A., Popović, Z., Style-based Inverse Kinematics, *ACM Transactions on Graphics (Proc. of SIGGRAPH 2004)*, pp. 522-531, 2004.

[Hot33a] Hotelling, H., Analysis of a complex of statistical variables into principal components, *Journal of Educational Psychology*, vol.24, pp. 417-441, 498-520, 1933.

[Jol02a] Jolliffe, I.T., *Principal Component Analysis*, 2nd edition, Springer, 2002.

[Law03a] Lawrence, N.D., Gaussian Process Latent Variable Models for Visualisation of High Dimensional Data, *Proc. of NIPS 2003*, 2004.

[Law05a] Lawrence, N.D., Probabilistic non-linear principal component analysis with Gaussian process latent variable models, *Journal of Machine Learning Research*, vol. 6, pp. 1783-1816, 2005.

[Law06a] Lawrence, N.D., Quiñonero Candela, J., Local distance preservation in the GP-LVM through back constraints, *Proc. of the 23rd International Conference on Machine Learning*, pp. 513-520, 2006.

[Pea01a] Pearson, K., On Lines and Planes of Closest Fit to Systems of Points in Space, *Philosophical Magazine*, vol.2, no.6, pp. 559-572, 1901.

[Ras06a] Rasmussen, C.E., Williams, C.K.I., *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[Urt06a] Urtasun, R., Fleet, D.J., and Fua, P., 3D people tracking with Gaussian process dynamical models, *2006 IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 238-245, 2006.

[Wan05a] Wang, J. M., Fleet, D. J., Hertzmann, A., Gaussian Process Dynamical Models, *Proc. of NIPS 2005*, pp. 1441-1448, 2006.

[Wan07a] Wang, J. M., Fleet, D. J., Hertzmann, A., Gaussian Process Dynamical Models for Human Motion, to appear in *IEEE Trans. on Pattern Analysis and Machine Intelligence*