# Modeling Human Locomotion with Topologically Constrained Latent Variable Models

Raquel Urtasun[1], David J. Fleet[2], and Neil D. Lawrence[3]

[1] Massachusetts Institute of Technology, Cambridge, MA 02139 USA
[2] University of Toronto, Canada M5S 3H5
[3] School of Computer Science, University of Manchester, M13 9PL, U.K.

**Abstract.** Learned, activity-specific motion models are useful for human pose and motion estimation. Nevertheless, while the use of activity-specific models simplifies monocular tracking, it leaves open the larger issues of how one learns models for multiple activities or stylistic variations, and how such models can be combined with natural transitions between activities. This paper extends the Gaussian process latent variable model (GP-LVM) to address some of these issues. We introduce a new approach to constraining the latent space that we refer to as the locally-linear Gaussian process latent variable model (LL-GPLVM). The LL-GPLVM allows for an explicit prior over the latent configurations that aims to preserve local topological structure in the training data. We reduce the computational complexity of the GPLVM by adapting sparse Gaussian process regression methods to the GP-LVM. By incorporating sparsification, dynamics and back-constraints within the LL-GPLVM we develop a general framework for learning smooth latent models of different activities within a shared latent space, allowing the learning of specific topologies and transitions between different activities.

## 1 Introduction

Modeling human motion is important for computer vision, computer graphics, orthopedics, sports and the entertainment industry (e.g., movies, computer games). In computer vision, for example, pose and motion models help resolve ambiguities in monocular people tracking. Nevertheless, learning human motion models is challenging since we must learn models from relatively sparse training data while avoiding overfitting, and the models must generalize to previously unobserved styles. One common approach has been to focus on *activity-specific* models [4,12,19,26]. This greatly simplifies learning, but leaves open the larger issues of how one learns models for a wide range of activities and stylistic variations, and how such models are combined with natural transitions from one activity to another [4,26]. To address these problems, this paper introduces a new form of Gaussian process latent variable model (GPLVM) [10] for learning models of different activities within a shared latent space, along with transitions between activities.

A. Elgammal et al. (Eds.): Human Motion 2007, LNCS 4814, pp. 104–118, 2007.

The GPLVM and the Gaussian process dynamical model (GPDM) have been used to learn generative models for human motion, including walking, running, and baseball pitching, proving useful for people tracking and computer animation [5,14,24,27,26,28]. Nonetheless, when there are large stylistic variations or different activities, the Gaussian process framework can produce models that are not smooth, fail to generalize well to nearby motions, and are therefore unsuitable for tracking and simulation [25].

One major problem lies with the formulation of the GPLVM. In its standard form the GPLVM ensures that the mapping from the latent space to the pose space is smooth, but not the map from pose space to latent space. That is, nearby latent positions will generate similar poses, but similar poses do not necessarily map to nearby latent positions. It is interesting to contrast this behaviour with other methods for nonlinear dimensionality reduction, such LLE [18], which aim to preserve the local topological structure of the training data. Unfortunately such methods for embedding do not provide generative models.

To preserve topological structure, the back-constrained GPLVM [8], introduces smoothness by constraining the latent positions to be a smooth function of the data space. The first goal of this paper is to generalize this approach with the formulation of a generative latent variable model whose prior, like the objective function used for LLE, aim to preserve local topological structure of the training data. We refer to this model as the locally-linear Gaussian process latent variable model (LL-GPLVM).

The second goal of this paper is to explore the ways in which one can constrain learning to produce topologically meaningful latent models. Most existing methods for learning motion models ignore useful prior information about human motion, such as the cyclic nature of locomotion, the physical laws at play, and the limited ways people transition between activities. Wang et al. [29] proposed a multifactor model for learning distributions of styles of human motion, parametrizing the space of human motion styles by a small number of low-dimensional factors, such as identity and gait. Their multifactor model can be viewed as a special case of the GPLVM, where the covariance function of the GP is a product of covariance functions for the individual factors. Here we take a complementary approach, incorporating prior knowledge about different activities and transitions between them within the LL-GPLVM and the back-constrained GPLVM. Importantly, transitions between different activities do *not* have to be present in the training data to be learned.

Finally, the application of the GPLVM has also been limited to small training sets because of its computational complexity, $O\left(N^3\right)$, where $N$ is the number of data points. One way to reduce this computational burden is to use the informative vector machine [9] to obtain a sparse representation; however, by using just a subset of the data, this approach ignores valuable training data, and is not guarrenteed to converge. We review the use of sparse Gaussian process regression in the GPLVM [11] in the context of human motion data. This results in a more effective algorithm, which converges to a generative model that depends on the entire data set.

## 2 Related Work

It has long been accepted that useful representations for visual analysis should capture the intrinsic structure of the data. This is the motivation to separate style and content with multilinear models for example [23,6,29]. Such models embody the tacit assumptions that there are a number of somewhat independent causes generating the data, and that these should be represented with separate dimensions. More generally, to learn such models, it is also often important to exploit all available prior knowledge about the domain.

Elgammal et al. [4] showed that for walking, nearby poses on a single person's gait are more similar than poses from other peoples'. As a consecuence, even for a single activity like walking, it can be challenging to learn a coherent model for multiple people, in which the motions are aligned and can be interpolated smoothly to generate new plausible styles. This becomes more difficult with multiple activities, where we wish to interpolate over different people's poses if their style is sufficiently similar, or to transition from one activity to another.

Perhaps the simplest approach to the general modeling problem is to use non-parametric models (e.g., [1,7,19]). This approach was used successfully for multi-activity character animation and tracking, but it does not easily generalize to nearby activities and styles, so one must have an enormous training corpus.

Classical multilinear models [23,6] produce style-content separation but are not suitable for many types of motions such as those with cyclic behaviour or nonlinearities [4]. Moreover, they have not been extended to handle transitions between activities. Elgammal et al. [4] learn a nonlinear model with stylistic variation (multiple people) for walking by first building individual models, and then using nonlinear regression to align the manifolds to build a unified model. While evocative, it is not clear whether this piecemeal approach will scale to more complex motions, multiple motions, or to many people. Wang et al. [29] propose a multifactor model for learning distributions of styles of human locomotion (walking, running and striding) but they do not explicitly model transitions. Somewhat smooth transitions can be achieved by linearly inerpolating the style factors.

Switching LDS [13,15] and HMMs [3] can represent multiple motions and diverse styles. Switching models have attractive properties that are somewhat complementary to the GP-LVM, but at present they require large amount of training data. Smoothness of the global model is another issue, as is the intractability of learning. In practice, it is sometimes argued that current switching models do not generalize well beyond the training data.

GP models [26,14,28] have proven very effective when dealing with a single motion type. Nevertheless, as discussed above, they have problems when dealing with multiple motions and different styles [25]. The aim of this paper is to discuss recently developed formulations that help address these issues.

## 3 Gaussian Process Latent Variable Models (GP-LVM)

We begin with a brief review of the GP-LVM. The GP-LVM represents a high-dimensional data set, $\mathbf{Y}$, through a low dimensional latent space, $\mathbf{X}$, and a Gaussian process mapping from the latent space to the data space. Let $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N]^T$ be a matrix in which each row is a single training datum, $\mathbf{y}_i \in \Re^D$. Let $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^T$ denote the matrix whose rows represent the corresponding positions in latent space, $\mathbf{x}_i \in \Re^d$. The GPLVM is a generative model of the data

$$\mathbf{y}_t = \sum_j \mathbf{b}_j f_j(\mathbf{x}_t) + \mathbf{n}_{y,t} \tag{1}$$

for weights $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, ...]$, basis functions $f_j$ and additive zero-mean white Gaussian noise $\mathbf{n}_{y,t}$. Given a covariance function for the Gaussian process, $k_Y(\mathbf{x}, \mathbf{x}')$, the likelihood of the data given the latent positions is,

$$p(\mathbf{Y} \mid \mathbf{X}, \bar{\beta}) \;=\; \frac{1}{\sqrt{(2\pi)^{ND}|\mathbf{K}_Y|^D}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{Y}^T\right)\right), \tag{2}$$

where $\mathbf{K}_Y$ is known as the kernel matrix, and $\bar{\beta}$ is a vector of kernel hyperparameters. The elements of the kernel matrix are defined by the covariance function, $(\mathbf{K}_Y)_{i,j} = k_Y(\mathbf{x}_i, \mathbf{x}_j)$. A common choice is the radial basis function (RBF), $k_Y(\mathbf{x}, \mathbf{x}') = \beta_1 \exp(-\frac{\beta_2}{2}||\mathbf{x} - \mathbf{x}'||^2) + \frac{\delta_{\mathbf{x},\mathbf{x}'}}{\beta_3}$, where $\bar{\beta} = \{\beta_1, \beta_2, ...\}$ are the kernel hyperparameters that determine the output variance, the RBF support width, and the variance of the additive noise. Learning in the GP-LVM consists of maximizing (2) with respect to the latent space configuration, $\mathbf{X}$, and the hyperparameters, $\bar{\beta}$.

When one has time-series data, $\mathbf{Y}$ represents a sequence of observations, and it is natural to augment the GP-LVM with an explicit dynamical model. For example, the Gaussian Process Dynamical Model (GPDM) models the latent sequence as a latent stochastic process with a Gaussian process prior [28], i.e.,

$$p(\mathbf{X} \mid \bar{\alpha}) \;=\; \frac{p(\mathbf{x}_1)}{\sqrt{(2\pi)^{(N-1)d}|\mathbf{K}_X|^d}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_X^{-1}\mathbf{X}_{out}\mathbf{X}_{out}^T\right)\right) \tag{3}$$

where $\mathbf{X}_{out} = [\mathbf{x}_2, ..., \mathbf{x}_N]^T$, $\mathbf{K}_X \in \Re^{(N-1)\times(N-1)}$ is the kernel matrix constructed from $\mathbf{X}_{in} = [\mathbf{x}_1, ..., \mathbf{x}_{N-1}]$, $\mathbf{x}_1$ is given an isotropic Gaussian prior and $\bar{\alpha}$ are the kernel hyperparameters. Below we use an RBF kernel for $\mathbf{K}_X$. Like the GP-LVM the GPDM provides a generative model for the data, but additionally it provides one for the dynamics. One can therefore predict future observation sequences given past observations, and simulate new sequences.

## 4 Sparse Approximations

By exploiting a sparse approximation to the full Gaussian process it is usually possible to reduce the computational complexity from an often prohibitive

$O\left(N^3\right)$ to a more manageable $O\left(k^2 N\right)$, where $k$ is the number of points retained in the sparse representation [11]. A large body of recent work has been focussed on approximating the covariance function with a low rank approximation [9,16,21]. All approximations involve augmenting the function values at the training points, $\mathbf{F} \in \Re^{N \times d}$, with $\mathbf{F} = [\mathbf{f}_1, ..., \mathbf{f}_N]^T$ and the function values at the test points, $\mathbf{F}_* \in \Re^{\infty \times d}$, by an additional set of variables, $\mathbf{U} \in \Re^{k \times d}$, called inducing variables [16]. The number of these variables, $k$, can be specified by the user.

The factorisation of the likelihood across the columns of $\mathbf{Y}$ allows us to focus on one column of $\mathbf{F}$ without loss of generality. We therefore consider function values at $\mathbf{f} \in \Re^{N \times 1}$, $\mathbf{f}_* \in \Re^{\infty \times 1}$ and $\mathbf{u} \in \Re^{k \times 1}$. These variables are considered to be jointly Gaussian distributed with $\mathbf{f}$ and $\mathbf{f}_*$ such that

$$p\left(\mathbf{f}, \mathbf{f}_*\right) = \int p\left(\mathbf{f}, \mathbf{f}_* | \mathbf{u}\right) p\left(\mathbf{u}\right) d\mathbf{u},$$

where the prior distribution over the inducing variables is given by a Gaussian process,

$$p\left(\mathbf{u}\right) = N\left(\mathbf{u} | \mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right),$$

with a covariance function given by $\mathbf{K}_{\mathbf{u},\mathbf{u}}$. This covariance is constructed on a set of inputs $\mathbf{X}_\mathbf{u}$ which may or may not be a subset of $\mathbf{X}$. Assume that the variables associated with the training data, $\mathbf{f}$, are conditionally independent of those associated with the test data, $\mathbf{f}_*$, given the inducing variables, $\mathbf{u}$[16]

$$p\left(\mathbf{f}, \mathbf{f}_*, \mathbf{u}\right) = p\left(\mathbf{f} | \mathbf{u}\right) p\left(\mathbf{f}_* | \mathbf{u}\right) p\left(\mathbf{u}\right),$$

where

$$p\left(\mathbf{f} | \mathbf{u}\right) = N\left(\mathbf{f} | \mathbf{K}_{\mathbf{f},\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, \mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u},\mathbf{f}}\right)$$

is the training conditional and

$$p\left(\mathbf{f}_* | \mathbf{u}\right) = N\left(\mathbf{f}_* | \mathbf{K}_{*,\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u},*}\right)$$

is the test conditional. $\mathbf{K}_{\mathbf{f},\mathbf{u}}$ is the covariance function computed between the training inputs, $\mathbf{X}$, and the inducing variables, $\mathbf{X}_\mathbf{u}$, $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ is the symmetric covariance between the training inputs, $\mathbf{K}_{*,\mathbf{u}}$ is the covariance function between the test inputs and the inducing variables and $\mathbf{K}_{*,*}$ is the symmetric covariance function the test inputs. This decomposition does not in itself entail any approximations: the approximations are introduced through assumptions about the form of these distributions. Here we consider the Fully Independence approximation (FITC) of Snelson and Ghahramani [21], where the training conditional is assumed to be

$$q\left(\mathbf{f} | \mathbf{u}\right) = N\left(\mathbf{f}_{(j)} | \mathbf{K}_{\mathbf{f},\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, diag\left(\mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u},\mathbf{f}}\right)\right).$$

For more details and other approximations we refer the reader to [11].

The smaller the number of inducing variables, the greater the computational and memory savings. Furthermore, estimation of a large number of inducing

variables may come with a risk of overfitting. However, a small number of inducing variables can result in poor reconstruction. In general the optimal amount of inducing variables is a function of the variation of the training data. If there is small variability in the training data (e.g. few persons with similar styles), a small amount of inducing variables is sufficient for accurate reconstruction. On the other hand, if the test data is very different from the training data, one has to choose a small number of inducing variables to avoid overfitting and generalize well to unseen styles. Unfortunately, there is no optimal way to automatically choose the number of inducing variables, and one has, for example, to use a validation set.

## 5 Top Down Imposition of Topology

The smooth mapping in the GP-LVM ensures that distant points in data space remain distant in latent space. However, as discussed in [8], the mapping in the opposite direction is not required to be smooth. While the GPDM may mitigate this effect, it often produces models that are neither smooth nor generalize well [26,28].

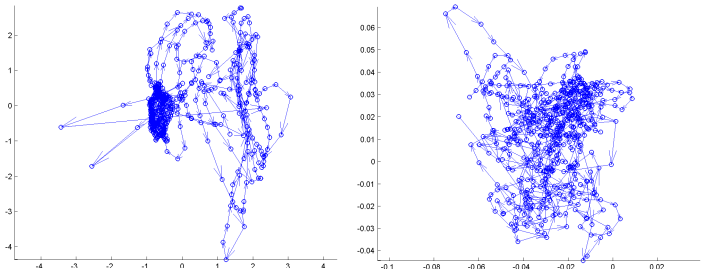
To help ensure smoother, well-behaved models, [8] suggest the use of *back-constraints*, wherein each point in the latent space is a smooth function of its corresponding point in data space, $x_{ij} = g_j(\mathbf{y}_i; \mathbf{a}_j)$, where $\{\mathbf{a}_j\}$ is the set of parameters of the mapping. Optimisation proceeds by substituting each $x_{ij}$ in (2) with $g_j(\mathbf{y}_i; \mathbf{a}_j)$ and maximizing the likelihood with respect to $\{\mathbf{a}_j\}$. This approach is known as the back-constrained GP-LVM.

Nevertheless, when learning human motion data with large stylistic variations or different motions, neither GPDM nor back-constrainted GP-LVM produce smooth models that generalize well. For example, Fig. 1(a) shows a GPDM learned from a training set comprising 9 walks and 10 runs. Compared to the models in Fig. 3 learned from the same training data, the GPDM (Fig. 1(a)) and the back-constrained GPDM (Fig. 1(a)) do not generalize to new runs and walks well, nor do they produce realistic looking simulations.

In this paper we first consider an alternative approach to the hard constraints on the latent space arising from $g_j(\mathbf{y}_i; \mathbf{a}_j)$. We introduce topological constraints through a prior distribution in the latent space, based on a neighborhood structure learn through a generalized local linear embedding (LLE) [18]. We then show how to incorporate domain-specific prior knowledge. This allows us to develop motion models with specific topologies that incorporate different activities within a single latent space and transitions between them.

### 5.1 Locally Linear GP-LVM

The locally linear embedding (LLE) [18] preserves topological constraints by finding a representation based on reconstruction in a low dimensional space with an optimized set of local weightings. Here we show how the LLE objective can be combined with the GP-LVM, yielding a *locally linear GP-LVM* (LL-GPLVM).



(a) (b)

**Fig. 1.** When training data contain large stylistic variations and multiple motions, the generic GPDM (a) and the back-constrained GPDM (b) do not produce useful models. The latent models here were learned with the same training data as used to learn those Fig. 3. Simulations of both models here do not look realistic.

The locally linear embedding assumes that each data point and its neighbors lie on, or close to, a locally linear patch on the data manifold. The local geometry of these patches can then be characterized by linear coefficients that reconstruct each data point from its neighbors. This is done in a three step procedure: (1) the $K$ nearest neighbors, $\{\mathbf{y}_j\}_{j\in\eta_i}$, of each point, $\mathbf{y}_i$, are computed using Euclidean distance in the input space, $d_{ij} = ||\mathbf{y}_i - \mathbf{y}_j||^2$; (2) the weights $\mathbf{w} = \{w_{ij}\}$ that best reconstruct each data point from its neighbors are obtained by minimizing $\Phi(\mathbf{w}) = \sum_{i=1}^{N} ||\mathbf{y}_i - \sum_{j\in\eta_i} w_{ij}\mathbf{y}_j||^2$; and (3) the latent positions $\mathbf{x}_i$ best reconstructed by the weights $w_{ij}$ are computed by minimizing $\Phi(\mathbf{X}) = \sum_{i=1}^{N} ||\mathbf{x}_i - \sum_{j\in\eta_i} w_{ij}\mathbf{x}_j||^2$. In the LLE, the weight matrix $\mathbf{w} = [w_{ij}]$, is sparse (only a small number of neighbors is used), and the two minimizations can be computed in close form.

**Locally Linear GP-LVM.** The LLE energy function is a function of the latent positions and can be interpreted, for a given set of weights $\mathbf{w}$, as a prior over latent configurations that forces each latent point to be locally reconstruct by its neighbors[1]. $p(\mathbf{X}|\mathbf{w}) = \frac{1}{Z}\exp\left\{-\frac{1}{\sigma^2}\Phi(\mathbf{X})\right\}$ , where $Z$ is a normalization constant, and $\sigma^2$ represents a global scaling of the prior. Following [18], we first compute the neighbors based on Euclidean distance. For each training point $\mathbf{y}_i$, we then compute the weights in closed form by solving, $\forall j \in \eta_i$, the following system,

$$\sum_k C_{kj}^{sim} w_{ij}^{sim} = 1, \quad \text{where} \quad C_{kj}^{sim} = \begin{cases} (\mathbf{y}_i - \mathbf{y}_k)^T(\mathbf{y}_i - \mathbf{y}_j), & \text{if } j,k \in \eta_i \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Once the weights are computed, they are rescaled so that $\sum_j w_{ij} = 1$.

[1] Strictly speaking this is not a proper prior as it is conditioned on the weights, and the weights depend on the training data.

Learning the locally linear GP-LVM is then equivalent to minimizing the negative log posterior of the model, that is up to an additive constant equal to [2]

$$\begin{aligned}\mathcal{L}_{\mathcal{S}} &= \log\, p(\mathbf{Y}|\mathbf{X},\hat{\beta})\, p(\hat{\beta})\, p(\mathbf{X}|\mathbf{w}) \\ &= \frac{D}{2}\ln|\mathbf{K}_Y| + \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{Y}^T\right) + \sum_i \ln\beta_i + \frac{1}{\sigma^2}\sum_{i=1}^{N}\|\mathbf{x}_i - \sum_{j=1}^{N} w_{ij}\mathbf{x}_j\|^2 \quad (5)\end{aligned}$$

Fig. 2 (a) shows a model compose of 2 walks and 2 runs learned with the Locally linear GPDM. Note how smooth the latent trajectories are.

In what follows we generalize the top-down imposition of topology strategies (i.e. back-constraints and locally linear GP-LVM) to incorporate domain specific prior knowledge.

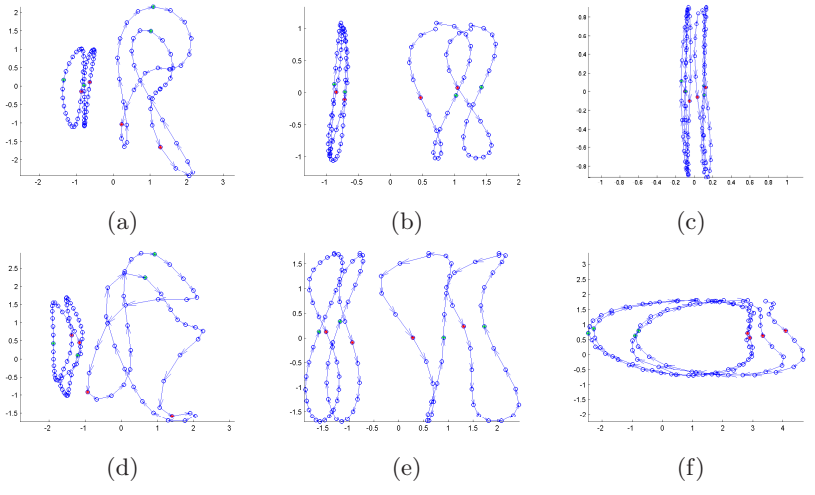



**Fig. 2.** First two dimensions of models learned using (a) LL-GPDM (b) LL-GPDM with topology (c) LL-GPDM with topology and transitions. (d) Back-constrained GPDM with and RBF mapping. (e) GPDM with topology through backconstraints. (f) GPDM with backconstraints for the topology and transitions. For the models using topology, the cyclic structure is imposed in the last 2 dimensions. The two types of transition points (left and right leg contact points) are shown in red and green, and are used as prior knowledge in (c,f).

## 6 Reflecting Knowledge in Latent Space Structure

A problem for modelling human motion data is the sparsity of the data relative to the diversity of naturally plausible motions. For example, while we might have a

[2] When learning a locally linear GPDM, the dynamics and the locally linear prior are combined as a product of potentials. The objective function becomes $\mathcal{L}_{\mathcal{S}} + \frac{d}{2}\ln|\mathbf{K}_X| + \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_X^{-1}\mathbf{X}_{out}\mathbf{X}_{out}^T\right) + \sum_i \ln\alpha_i$, with $\mathcal{L}_{\mathcal{S}}$ defined as in (5).

data set comprising different motions, such as runs, walks *etc.*, the data may not contain transitions between motions. In practice however, we know that these motions will be broadly cyclic and that transitions can only physically occur at specific points in the cycle. How can we encourage our model to respect such topological constraints which arise from prior knowledge?

We consider this problem both in the context of the functionally constrained GP-LVM and the locally linear GP-LVM introduced above. We show how one can adjust the distance metric used in the locally linear embedding to better reflect different types of prior knowledge (such as the location of possible transition points). We also show how one can define similarity measures for use with the functionally constrained GP-LVM. Both these approaches force the latent space to construct a representation that reflects our prior knowledge.

### 6.1 Prior Knowledge with Back Constraints

As we mentioned in Section 3, back constraints involve a mapping from data space to latent space. If the mapping is smooth the constraints forces points that are close in data space to be close in latent space. The back-constraint functions are used to explicitly constrain the latent positions associated with training points. One possible mapping is a kernel-based regression model, where a regression, on a kernel induced feature space, provides the mapping, $x_{ij} = \sum_{m=1}^{N} a_{jm} k(\mathbf{y}_n, \mathbf{y}_m)$. Many kernels have interpretations as similarity measures [17]. In particular, any similarity measure that leads to a positive semi-definite matrix can be interpreted as a kernel. When learning the back-constrained GP-LVM, one needs to determine the hyperparameters of the kernel matrices (for the back-constraints and the covariance of the GP), as well as the weights of the mapping, $\{a_{jm}\}$. Following [8], we fixed the hyperparameters of the back-constraint's kernel matrix, and optimized over the remaining parameters. We extend the original formulation of back constraints by constructing similarity measures (i.e. kernels) to reflect prior knowledge. We consider two examples: the first involves transitions between activities; with the second we show how topological constraints can be placed on the form of the latent space.

**Similarity for Transitions.** To capture transitions between two motions, we wish to design a kernel that expresses strong similarity between points in the respective motions where transitions may occur. To express this similarity we first define an index on the frames of the motion sequence, $\{t_i\}_{i=1}^{N}$. We then define subsets of this set, $\{\hat{t}_i\}_{i=1}^{M}$, which represents frames where transitions are possible. For walking and running motions one might use two transition sets, one in which the left foot makes ground contact, and one for the right foot; ground contact can be automatically extracted as it coincides with non linearity in the dynamics [2]. For other motion types a similarity measure can be used to define transitions [7]. We can encourage transition points of different sequences to be proximal with the following kernel matrix for the back-constraint mapping:

$$k^{trans}(t_i, t_j) = \sum_m \sum_l \delta_{ml} k(t_i, \hat{t}_m) k(t_j, \hat{t}_l) \tag{6}$$

where $k(t_i, \hat{t}_l)$ is an RBF centered at $\hat{t}_l$, and $\delta_{ml} = 1$ if $\hat{t}_m$ and $\hat{t}_l$ are in the same set. The 'strength' of the encouragement is controlled by the support width of the RBF kernel.

**Combining Similarities.** One advantage of our framework is that kernel matrices can be combined in a principled manner to form new kernel matrices. Kernels can be multiplied (on an element by element basis) or added together. Multiplication of kernel-based similarity measures has, loosely speaking, an 'AND gate effect', i.e. both similarity measures must agree that an object is similar for their product to express similarity. Adding them produces more of an 'OR gate effect', i.e. if either representation expresses similarity the resulting measure will also express similarity.

**Topologically Constrained Latent Spaces.** We now consider kernels that encourage the latent space to have a particular topology. Specifically we are interested in suitable topologies for walking and running data. Because the data are approximately periodic, it seems appropriate to have a non-Cartesian topology. To this end one could extract the phase of the motion[3], $\phi$, and use it with a suitable similarity measure to encourage the latent points to response a non-Cartesian topology within a Cartesian space. As an illustrative example we consider a cylindrical topology within a three dimensional latent space by constraining two of the latent dimensions with phase. In particular to represent the cyclic motion we construct a distance function on the unit circle, where a latent point corresponding to phase $\phi$ is represented with the point $(\cos(\phi), \sin(\phi))$. A periodic mapping can be constructed from a kernel matrix as follows,

$$\begin{aligned} x_{n,1} = g_1^{cos}(\mathbf{y}_n, \phi_n) &= \sum_{m=1}^{N} a_m^{cos} k(\cos(\phi_n), \cos(\phi_m)) + a_0^{cos} \delta_{n,m}, \\ x_{n,2} = g_2^{sin}(\mathbf{y}_n, \phi_n) &= \sum_{m=1}^{N} a_m^{sin} k(\sin(\phi_n), \sin(\phi_m)) + a_0^{sin} \delta_{n,m}, \end{aligned}$$

where $k$ is an RBF kernel function, and $x_{n,i}$ is the $i^{th}$ coordinate of the $n^{th}$ latent point. These two mappings project onto two dimensions of the latent space, forcing them to have a periodic structure (which comes about through the dependence of the kernel with respect to cosine and sine of the phase). Fig. 2 (e) shows a model learned using GPDM with the last two dimensions constrained in this way (the third dimension is out of plane). The first dimension is constrained by an RBF mapping on the input space. Each dimension's kernel matrix can then be augmented by adding the transition similarity given in (6), resulting in the model depicted by Fig. 2 (f).

[3] The phase can be easily extracted from the data by Fourier analysis or by detecting key postures and interpolating the phases between them. Another idea, not further explored here, would be to optimize the GP-LVM with respect to the phase.

### 6.2 Prior Knowledge Through Local Linearities

We now turn to consider how one might incorporate prior knowledge in the locally linear GP-LVM framework. This is accomplished by replacing the local Euclidean distance measures presented in Section 5.1 with other similarity measures. The standard Euclidean distance measure leads to the covariance matrice given in (4). However, just as we developed similarity matrices above, we can also modify this covariance function to reflect our prior knowledge in the latent space.

**Covariance for Transitions.** To capture transitions in the latent model we define the elements for the covariance matrix as follows,

$$C_{ij}^{trans} = 1 - \left[\delta_{ij} \exp(-\zeta(t_i - t_j)^2)\right] \tag{7}$$

with $\zeta$ a constant, and $\delta_{ij} = 1$ if $t_i$ and $t_j$ are in the same set $\{\hat{t}_k\}_{k=1}^{M}$, and otherwise $\delta_{ij} = 0$. This covariance encourages the points at which transitions are physically possible to be close together in the latent space.

**Combining Covariances.** As above the covariance matrices can also be combined in a principled way. Covariances can be added or multiplied (on an element by element basis) to loosely speaking produce and 'OR' or 'AND' gate effect.

**Covariance for Topologies.** To force a cylindrical topology on the latent space, we can introduce covariances based on the phase, specifying different covariances for each latent dimensions. As before we construct a distance function in the unit circle, that takes into account the phase.

$$C_{j,k}^{cos} = \big(\cos(\phi_i) - \cos(\phi_{\eta_j})\big)\ \big(\cos(\phi_i) - \cos(\phi_{\eta_k})\big) \tag{8}$$

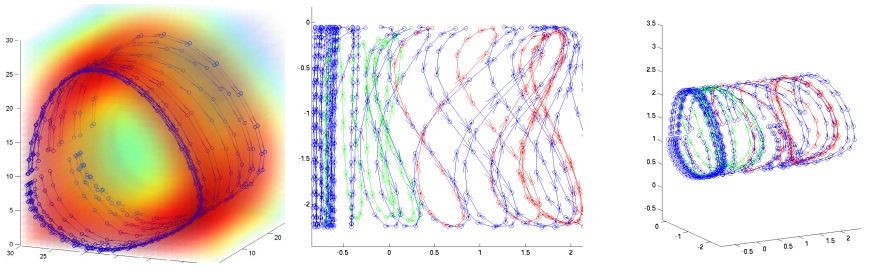
$$C_{j,k}^{sin} = \big(\sin(\phi_i) - \sin(\phi_{\eta_j})\big)\ \big(\sin(\phi_i) - \sin(\phi_{\eta_k})\big)\,, \tag{9}$$

The covariance for the remaining dimension is constructed as usual, based on Euclidean distance in the data space. In Fig. 2 (b) a GPDM is constrained in this way, and in Fig. 2 (c) the covariance is augmented with transitions.

Note that the use of different distance measures for each dimension of the latent space implies that the neighborhood and the weights in the locally linear prior will also be different for each dimension. One way of looking at it is that three different locally linear embeddings are developed for constructing the prior distribution. Each gives a prior distribution for a different dimension of the latent space.

### 6.3 Model Combination

The two sections above have shown how to incorporate prior knowledge in the GP-LVM by means of 1) local linearities and 2) backconstraints. In general, the latter converges faster but it is more intuitive to constraint the latent space with soft constraints. Both techniques are complementary and can be combined straightforwardly by including priors over some dimensions, and constraining the others through back-constraint mappings. Fig. 3 shows a model learned with the locally linear GPDM to impose smoothness and backconstraints for topology.



(a) (b) (c)

**Fig. 3.** Hybrid model learned using local linearities for the style and backconstraints for the content. The training data is composed of 9 walks and 10 runs performed by different subjects and speeds. (a) Likelihood for the reconstruction of the latent points (b) First two components and (c) 3D view of the latent trajectories for the training data in blue, and the automatically generated motions of Figs. 4 and 5 in green and red respectively.

### 6.4 Modeling Multiple Activities and Transitions Between Them

Once we know how to ensure that transition points are close together and that the latent structure has the topology we want, we still need to address two issues. How do we learn models that have very different dynamics? How will the model interpolate between the different dynamics? Our goal in this section is to show how latent models for different motions can be learned independently, but in a shared latent space that facilitates transitions between activities with different dynamics.

**Dynamics for Multiple Activities.** Let $\mathbf{Y} = [\mathbf{Y}_1^T, ..., \mathbf{Y}_M^T]^T$ denote training data for $M$ different activities. Each $\mathbf{Y}_m$ comprises several different motions. Let $\mathbf{X} = [\mathbf{X}_1^T, ..., \mathbf{X}_M^T]^T$ denote the corresponding latent positions. When dealing with multiple activities, a single dynamical model cannot cope with the complexity of the different dynamics. Instead, we consider a model where the dynamics of each activity are modeled independently, $p(\mathbf{X}) = \prod_{m=1}^{M} p(\mathbf{X}_m)$ . This approach has the advantage that a different kernel can be used for each acitivity. Another interpretation is that we have a block diagonal kernel matrix for the Gaussian process that governs the dynamics.

To enable interpolation between motions with different dynamics, we combined these independent dynamical models learned in the form of a mixture model. By interpolating between the components of the mixture distribution we can produce motions that gracefully transition between different styles and motion types (Figs. 4 and 5).

## 7 Results

We first evaluate the sparse GPLVM in the CMU Mocap dataset comprising 1613 poses from walking and running motions of subject 35. For learning we used a

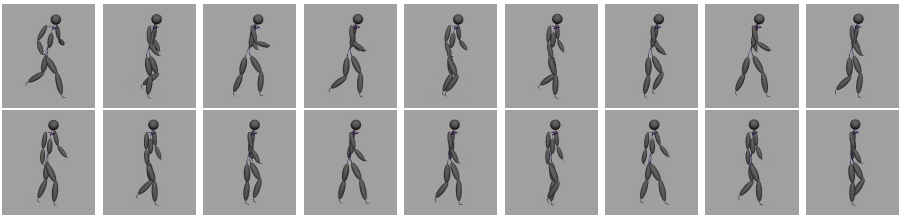


**Fig. 4. Transition from running to walking**: The system transitions from running to walking in a smooth and realistic way. The transition is encouraged by incorporating prior knowledge in the model. The latent trajectories are shown in green in Fig. 2 (b,c).

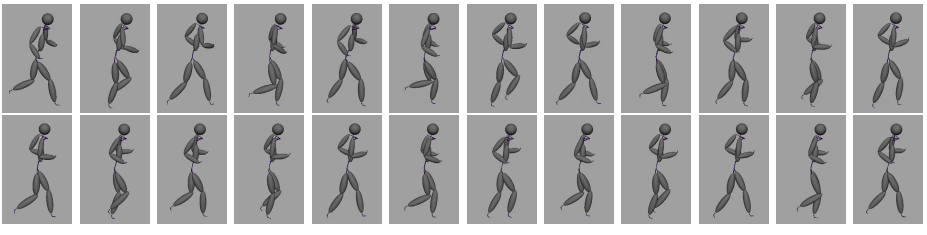


**Fig. 5. Different running styles and speeds**: The system is able to simulate a motion with considerably changes in speed and style. The latent trajectories are shown in red in Fig. 2 (b,c).

FITC approximation with 100 inducing points. However, rather than allowing these points to be moved freely, they were fixed to latent positions that were uniformily sub-sampled from the data. The models were also backconstrained to encourage smoothness. Following [22], we apply the model to two missing data problems, the first in which the right leg data was removed from a test sequence and the second in which the upper body was removed. A summary of the results is shown in Table 1, showing that that the GPLVM outperforms nearest neighbour, and that a latent space of dimension 3 is sufficient to model the data.

To illustrate how prior knowledge can impact the learned models, we consider a training set comprising 4 gait cycles (2 walks and 2 runs) performed by one

**Table 1.** Results from the missing data problem. Headings: L and B are the leg and body data sets. The error measure is the mean square error in the angle space. Methods are: NN: nearest neighbour, GPLVM (latent dimension): the GPLVM with different latent dimensions, $q$. Bold results are the best reported for a given column.

| Data | L | B |
|---|---|---|
| GPLVM ($q = 3$) | 3.40 | **2.49** |
| GPLVM ($q = 4$) | **3.38** | 2.72 |
| GPLVM ($q = 5$) | 4.25 | 2.78 |
| NN | 4.11 | 3.20 |

subject at different speeds. Figure 2 shows the latent spaces learned under different priors. All models are learned using two independent dynamical models, one for walking and one for running. Note how the phases are aligned when promoting a cylindrical topology, and how the LL-GPDM is smooth. Notice the difference between the LL-GPDM (Fig. 2 (c)) and the backconstrained GPDM (Fig. 2 (f)) when transition constraints are included. Both models ensure that the transition points (shown in red and green) are proximal.

Figure 3 (a) shows a hybrid model learned using LL-GPDM for smoothness, and back-constraints for topology. The larger training set comprises approximately one gait cycle from each of 9 walking and 10 running motions performed by different subjects at different speeds (3 km/h for walking, 6–12 km/h for running). Colors in Fig. 3 (a) represent the log variance of the GP as a function of latent position. Only points close to the surface of the cylinder generate poses with high certainty.

We now illustrate the model's ability to simulate different motions and transitions. Given an initial latent position $\mathbf{x}_0$, we generate new motions by sampling the mixture model, and using mean prediction for pose reconstruction. Choosing different initial conditions results in different simulations (see Fig. 3 (b-c), where the training data are shown in blue). For the first simulation (in green), the motion is initialized to a running pose, with a latent position not far from walking poses. The system transitions to walking quickly and quite naturally. The resulting animation is depicted in Fig. 4. For the second example (in red), we initialize the simulation to a latent position far from walking data. The system evolves to different running styles and speeds (Fig. 5). Note how the dynamics, and the strike length, change considerably during simulation.

## 8 Conclusions

In this paper we have proposed a general framework of probabilistic models that learn smooth latent variable models of different activities within a shared latent space. We have also introduced a principled way to include prior knowledge, that allow us to learn specific topologies and transitions between the different motions. Although we have learned models composed of walking and running, our framework is general, being applicable in any data sets where there is a large degree of prior knowledge for the problem domain, but the data availability is relatively sparse compared to its complexity.

## References

1. Arikan, O., Forsyth, D.: Interactive motion generation from examples. In: SIGGRAPH, pp. 483–490 (2002)
2. Bissacco, A.: Modeling and Learning Contact Dynamics in Human Motion. In: CVPR, pp. 421–428 (2005)
3. Brand, M., Hertzmann, A.: Style Machines. In: SIGGRAPH, pp. 183–192 (2000)
4. Elgammal, A., Lee, C.: Separating Style and Content on a Nonlinear Manifold. In: CVPR, vol. 1, pp. 478–485 (2004)

5. Grochow, K., Martin, S., Hertzmann, A., Popovic, Z.: Style-based inverse kinematics. In: SIGGRAPH, pp. 522–531 (2004)
6. Vasilescu, M.A.: Human Motion Signatures: Analysis, Synthesis. In: ICPR, pp. 456–460 (2002)
7. Kovar, L., Gleicher, M., Pighin, F.: Motion Graphs. In: SIGGRAPH, pp. 473–482 (2002)
8. Lawrence, N., Quinonero-Candela, J.: Local distance preservation in the GP-LVM through back constraints. In: ICML, pp. 513–520 (2006)
9. Lawrence, N.D., Seeger, M., Herbrich, R.: Fast sparse Gaussian process methods: The informative vector machine. In: NIPS, pp. 609–616 (2003)
10. Lawrence, N.D.: Gaussian Process Models for Visualisation of High Dimensional Data. In: NIPS (2004)
11. Lawrence, N.: Learning for larger datasets with the Gaussian process latent variable model. In: AISTATS (2007)
12. Li, R., Yang, M.H., Sclaroff, S., Tian, T.: Monocular Tracking of 3D Human Motion with a Coordinated Mixture of Factor Analyzers. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 137–150. Springer, Heidelberg (2006)
13. Li, Y., Wang, T., Shum, H.: Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis. In: SIGGRAPH, pp. 465–472 (2002)
14. Moon, K., Pavlovic, V.: Impact of Dynamics on Subspace Embedding and Tracking of Sequences. In: CVPR, pp. 198–205 (2006)
15. Pavlovic, J.M., Rehg, J., MacCormick, J.: Learning switching linear models of human motion. In: NIPS, pp. 981–987 (2000)
16. Quiñonero-Candela, J., Rasmussen, C.E.: A Unifying View of Sparse Approximate Gaussian Process Regression. In: JMLR, vol. 6, pp. 1939–1959 (2006)
17. Rasmussen, C.E., Williams, C.K.: Gaussian Process for Machine Learning. MIT Press, Cambridge (2006)
18. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)
19. Sidenbladh, H., Black, M.J., Sigal, L.: Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In: Tistarelli, M., Bigun, J., Jain, A.K. (eds.) ECCV 2002. LNCS, pp. 784–800. Springer, Heidelberg (2002)
20. Sminchisescu, C., Jepson, A.: Generative Modeling for Continuous Non-Linearly Embedded Visual Inference. In: ICML, pp. 96–103 (2004)
21. Snelson, E., Ghahramani, Z.: Sparse Gaussian processes using pseudo-inputs. In: NIPS, pp. 1257–1264 (2006)
22. Taylor, G.W., Hinton, G.E., Roweis, S.: Modeling human motion using binary latent variables. In: NIPS, pp. 1345–1352 (2007)
23. Tenenbaum, J., de Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
24. Tian, T., Li, R., Sclaroff, S.: Articulated Pose Estimation in a Learned Smooth Space of Feasible Solutions. In: CVPR Learning Workshop (2005)
25. Urtasun, R.: Motion Models for Robust 3D Human Body Tracking. PhD thesis, EPFL (2006)
26. Urtasun, R., Fleet, D.J., Fua, P.: 3d people tracking with gaussian process dynamical models. In: CVPR, vol. 1, pp. 238–245 (2006)
27. Urtasun, R., Fleet, D.J., Hertzman, A., Fua, P.: Priors for people tracking from small training sets. In: ICCV, pp. 403–410 (2005)
28. Wang, J., Fleet, D.J., Hertzman, A.: Gaussian Process dynamical models. In: NIPS, pp. 1441–1448 (2005)
29. Wang, J., Fleet, D.J., Hertzman, A.: Multifactor Gaussian Process Models for Style-Content Separation. In: ICML, pp. 975–982 (2007)