International Conference on Machine Learning, Oregon, 2007 (to appear)

Multifactor Gaussian Process Models for Style-Content Separation

Jack M. Wang JMWANG@DGP.TORONTO.EDU
David J. Fleet FLEET@CS.TORONTO.EDU
Aaron Hertzmann HERTZMAN@DGP.TORONTO.EDU
Department of Computer Science, University of Toronto, Toronto, ON M5S 3G4 Canada

Abstract

We introduce models for density estimation with multiple, hidden, continuous *factors*. In particular, we propose a generalization of multilinear models using nonlinear basis functions. By marginalizing over the weights, we obtain a multifactor form of the Gaussian process latent variable model. In this model, each factor is kernelized independently, allowing nonlinear mappings from any particular factor to the data. We learn models for human locomotion data, in which each pose is generated by factors representing the person's identity, gait, and the current state of motion. We demonstrate our approach using time-series prediction, and by synthesizing novel animation from the model.

1. Introduction

Using prior models of human motion to constrain the inference of 3D pose sequences is a popular approach to improve monocular people tracking, as well as to simplify the process of character animation. The availability of motion capture devices in recent years enables such models to be learned from data, and learning models that generalize well to novel motions has become a major challenge.

One of the main difficulties in this domain is that the training data and test data typically come from related but distinct distributions. For example, we would often like to learn a prior model of locomotion from the motion capture data of a few individuals performing a few gaits (i.e., walking and running). Such a prior model could then be used to track a new individual or to generate plausible animations of a related, but new gait not included in the training database. Due to the natural variations in how different individuals perform different gaits — which we broadly refer to as *style* — learning a model that can represent and generalize to the space of human motions is not straightforward. One approach is to learn a single model from all training data, without regard of our knowledge about their style. However, this can lead to unrealistic models that either average together all styles of motion, or else amount to a mixture model of styles. Neither approach can be expected to handle data from new styles well. Nonetheless, it has long been observed that interpolating and extrapolating motion capture data yields plausible new motions, and it is reasonable to attempt building motion models that can generalize in style.

This paper introduces a multifactor model for learning distributions of styles of human motion. We parameterize the space of human motion styles by a small number of low-dimensional *factors*, such as identity and gait, where the dependence on each individual factor may be nonlinear. This parameterization is learned in a semi-supervised manner from a collection of example motions with different styles. Given a new motion, identifying its stylistic factors defines that motion's style-specific distribution.

Our multifactor Gaussian process model can be viewed as a special class of Gaussian process latent variable model (GPLVM) (Lawrence, 2005). As in the GPLVM, we marginalize out the weights in the generative model, and optimize the latent variables that correspond to the different factors in the model. If used with linear factors, the complete model amounts to a Bayesian generalization of multilinear models (De Lathauwer et al., 2000; Vasilescu & Terzopoulos, 2002). We also incorporate latent-space dynamics, and show that the use of the multifactor model improves time-series prediction results on human motion.

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

1.1. Background

The problem of style-content separation — modeling the interaction of multiple factors — was introduced by Tenenbaum and Freeman (2000). They employed a bilinear model, in which hidden "style" and "content" variables are multiplied along with a set of weights to produce observations; their algorithm was used to model variations in images of human faces and in typefaces. Identifying which variables correspond to "style" or "content" is problem-dependent, and somewhat arbitrary.

The natural generalization of the bilinear model when more than two factors are present is the multilinear model. Multilinear factorizations have been used to model images of faces (Vasilescu & Terzopoulos, 2002), 3D face geometry (Vlasic et al., 2005), motion capture sequences (Vasilescu, 2002), and texture and reflectance (Vasilescu & Terzopoulos, 2004). These models are multilinear in the factors, but linear with respect to any single factor. Li et al. (2005) proposed a multifactor generalization of kernel principal components analysis (PCA). It is complementary to the model proposed here, as they kernelize the outputs while we kernelize the factors.

The main application in this paper is learning models of human poses and motions. Perhaps the simplest approach to generating motion is to interpolate example poses (Rose et al., 2001) or motion capture sequences (Rose et al., 1998), assuming that all examples are labeled with style parameters. Independent components analysis can be applied to sequences to obtain a linear style-space of sequences (Shapiro et al., 2006).

A few methods for nonlinear style-content separation of human pose and motion also exist. The style machines model (Brand & Hertzmann, 2000) learns a linear space of style-specific hidden Markov models for different individuals. This method is limited to two factors, and must represent poses with a discrete state model plus temporal smoothing. Elgammal and Lee (2004) learn a nonlinear manifold and a two-factor mapping to pose and silhouette data in a least-squares setting.

More recently, several researchers have used the GPLVM (Lawrence, 2005) to model human poses (Grochow et al., 2004; Urtasun et al., 2005). Given a set of high-dimensional training poses, the GPLVM provides a set of corresponding low-dimensional latent coordinates, along with a Gaussian process mapping from latent coordinates to pose observations. The mapping is in general nonlinear, and gives rise to a joint distribution over new data and the corresponding latent coordinates. The Gaussian process dynamical model (GPDM) (Wang et al., 2006) extends the GPLVM by including a dynamical model on the low-dimensional latent space. It thereby models time-series data for a single individual, but does not generalize well to multiple styles or activities. This paper builds on these two models with the inclusion of factors to represent variation in gait and across individuals.

2. Multifactor Gaussian Processes

The model we use is a probabilistic latent variable model, involving a low-dimensional latent space of hidden factors describing style and content, and a mapping to a high-dimensional observation space. In this section, we introduce the multifactor Gaussian process (GP) mapping that lies at the core of our approach. We will assume for now that the inputs are known, and only consider one-dimensional outputs. In Section 3, we describe how to learn the model in an unsupervised fashion, and apply the model to motion capture data, in which each observation is a high-dimensional human body pose associated with a particular person and a specific gait.

2.1. Gaussian Processes

We begin by reviewing GP regression, using the "weight-space" view (Rasmussen & Williams, 2006). Suppose we have a one-dimensional function $y = g(\mathbf{x})$ of input vector $\mathbf{x}$, defined as a linear combination of J basis functions $\phi_j(\mathbf{x})$:

$$y = g(\mathbf{x}) = \sum_{j=1}^{J} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \mathbf{\Phi}(\mathbf{x}), \qquad (1)$$

where the vector $\mathbf{\Phi}(\mathbf{x}) = [\phi_1(\mathbf{x}), ..., \phi_J(\mathbf{x})]^T$ stacks the basis functions. Furthermore, we assume a weight decay prior: $\mathbf{w} \sim \mathcal{N}(0; \mathbf{I})$. Since the outputs y are a linear function of the weights, the outputs are also Gaussian. In particular, given known inputs $\mathbf{x}$ and $\mathbf{x}'$, the mean and covariance of their outputs y and y' are:

$$\begin{aligned} \mu(\mathbf{x}) &\equiv E[y] = E[\mathbf{w}^T \mathbf{\Phi}(\mathbf{x})] = 0, & (2) \\ k(\mathbf{x}, \mathbf{x}') &\equiv E[yy'] = E[(\mathbf{w}^T \mathbf{\Phi}(\mathbf{x}))(\mathbf{w}^T \mathbf{\Phi}(\mathbf{x}'))] \\ &= \mathbf{\Phi}(\mathbf{x})^T \mathbf{\Phi}(\mathbf{x}'), & (3) \end{aligned}$$

since $E[\mathbf{w}] = 0$ and $E[\mathbf{w}\mathbf{w}^T] = \mathbf{I}$. The functions $\mu(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$ are referred to as the mean function and kernel function, respectively. If we choose linear basis functions (i.e., $\mathbf{\Phi}(\mathbf{x}) = \mathbf{x}$), then the kernel function is quadratic: $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$. It can be shown that, with appropriate choice of Gaussian basis functions for

$\phi_j(\mathbf{x})$, the kernel function becomes the "RBF kernel":

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{\gamma}{2}||\mathbf{x} - \mathbf{x}'||^2). \quad (4)$$

Other assumptions about the form of g lead to different kernel functions.

Given N training pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$, the $N \times N$ kernel matrix $\mathbf{K}$ is defined such that $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. A Gaussian predictive distribution at a new input, $\tilde{\mathbf{x}}$ can then be derived i.e.,

$$\tilde{y} \,|\, \tilde{\mathbf{x}}, \mathcal{D} \;\sim\; \mathcal{N}(m(\tilde{\mathbf{x}}); \sigma^2(\tilde{\mathbf{x}})), \quad (5)$$

where

$$\begin{aligned} m(\mathbf{x}) &= [y_1, ..., y_N]\, \mathbf{K}^{-1}\mathbf{k}(\mathbf{x}), & (6)\\ \sigma^2(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}), & (7)\\ \mathbf{k}(\mathbf{x}) &= [k(\mathbf{x}, \mathbf{x}_1), ..., k(\mathbf{x}, \mathbf{x}_N)]^T. & (8) \end{aligned}$$

2.2. A Simple Two-Factor Model

Suppose now we wish to model different mappings for different styles. One way to do this is to add a latent "style" parameter. Accordingly, consider a regression problem with inputs $\mathbf{x}$ and style parameters $\mathbf{s} \in \mathbb{R}^S$. We define the following mapping, in which the output depends linearly on style:

$$\begin{aligned} y &= f(\mathbf{x}; \mathbf{s}) = \sum_{i=1}^{S} s_i g_i(\mathbf{x}) + \varepsilon \\ &= \sum_{i=1}^{S} s_i \mathbf{w}_i^T \mathbf{\Phi}(\mathbf{x}) + \varepsilon, \quad (9) \end{aligned}$$

where each $g_i(\mathbf{x})$ is a mapping with weight vector $\mathbf{w}_i$, and ε represents additive i.i.d. Gaussian noise with zero mean and variance β^{-1}. Fixing just the input $\mathbf{s}$ specializes the mapping to a specific style. If we hold fixed the input $\mathbf{x}$ and style $\mathbf{s}$, then, because ε and $\mathbf{w} \equiv [\mathbf{w}_1^T \ldots \mathbf{w}_S^T]^T$ are Gaussian, and $f(\mathbf{x}; \mathbf{s})$ is a linear function of $\mathbf{w}$, $f(\mathbf{x}; \mathbf{s})$ is also Gaussian. Given two sets of inputs $(\mathbf{x}, \mathbf{s})$ and $(\mathbf{x}', \mathbf{s}')$, this function has mean and covariance

$$\begin{aligned} E[y] &= \sum_i s_i E[\mathbf{w}_i]^T \mathbf{\Phi}(\mathbf{x}) + E[\varepsilon] = 0, & (10)\\ E[yy'] &= E[\left(\sum_{i=1}^{S} s_i g_i(\mathbf{x}) + \varepsilon\right)\left(\sum_{j=1}^{S} s'_j g_j(\mathbf{x}') + \varepsilon'\right)] \\ &= \sum_i s_i s'_i E[(\mathbf{w}_i^T \mathbf{\Phi}(\mathbf{x}))(\mathbf{w}_i^T \mathbf{\Phi}(\mathbf{x}'))] + E[\varepsilon\varepsilon'] \\ &= (\mathbf{s}^T\mathbf{s}')\, \mathbf{\Phi}(\mathbf{x})^T \mathbf{\Phi}(\mathbf{x}') + \beta^{-1}\delta. & (11) \end{aligned}$$

The term δ is 1 when y and y' are the same measurement, and zero otherwise.

The simple two-factor model with linear dependence on style and nonlinear dependence on content can therefore be expressed as a GP, with the Bayesian integration of the weights derived in closed form. As discussed below, the two-factor model can be generalized to greater numbers of factors, each of which may be linearly or nonlinearly related to the training data, holding the other factors fixed.

2.3. General Multifactor Models

In general, suppose we wish to model the effect of M factors $\mathcal{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)}\}$ on the output independently, then

$$\begin{aligned} y &= f(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)}) + \varepsilon \\ &= \mathbf{w}^T(\mathbf{\Phi}^{(1)} \otimes \cdots \otimes \mathbf{\Phi}^{(M)}) + \varepsilon, \quad (12) \end{aligned}$$

where $\mathbf{\Phi}^{(i)}$ is a basis column vector for factor $\mathbf{x}^{(i)}$, $\mathbf{w}$ is a weight vector, ε is as defined in the previous section, and $\otimes$ denotes the Kronecker product.[1] The lengths of $\mathbf{w}$ and $(\mathbf{\Phi}^{(1)} \otimes \cdots \otimes \mathbf{\Phi}^{(M)})$ are both equal to the product of the lengths of $\mathbf{\Phi}^{(i)}$'s.

As before, we assume a weight decay prior on $\mathbf{w}$. Hence, y is a GP with zero mean and covariance

$$\begin{aligned} k(\mathcal{X}, \mathcal{X}') &\equiv E[yy'] \\ &= E[\mathbf{w}^T(\mathbf{\Phi}^{(1)} \otimes \ldots)\mathbf{w}^T(\mathbf{\Phi}^{(1)'} \otimes \ldots)] + \beta^{-1}\delta \\ &= (\mathbf{\Phi}^{(1)} \otimes \ldots)^T(\mathbf{\Phi}^{(1)'} \otimes \ldots) + \beta^{-1}\delta \\ &= (\mathbf{\Phi}^{(1)^T}\mathbf{\Phi}^{(1)'}) \otimes \cdots \otimes (\mathbf{\Phi}^{(M)^T}\mathbf{\Phi}^{(M)'}) + \beta^{-1}\delta \\ &= \prod_{i=1}^{M} k_i(\mathbf{x}^{(i)}, \mathbf{x}^{(i)'}) + \beta^{-1}\delta, \quad (13) \end{aligned}$$

where $k_i(\mathbf{x}^{(i)}, \mathbf{x}^{(i)'}) = \mathbf{\Phi}^{(i)^T}\mathbf{\Phi}^{(i)'}$ is the kernel function for the i-th factor, $\mathbf{\Phi}^{(i)'}$ is a function of $\mathbf{x}^{(i)'}$. For example, the kernel function in (11) has two factors, with $k_1(\mathbf{s}, \mathbf{s}') = \mathbf{s}^T\mathbf{s}'$ and $k_2(\mathbf{x}, \mathbf{x}') = e^{-\frac{\gamma}{2}||\mathbf{x}-\mathbf{x}'||^2}$.

Given N training pairs $\mathcal{D} = \{(\mathcal{X}_i, y_i)\}_{i=1}^N$, the kernel matrix $\mathbf{K}$ for the resulting GP is defined in the usual way; i.e., $\mathbf{K}_{i,j} = k(\mathcal{X}_i, \mathcal{X}_j)$. The kernel product may also be written as the elementwise product of M kernel matrices, one for each factor,

$$\mathbf{K} \;=\; \mathbf{K}^{(1)} \circ \mathbf{K}^{(2)} \circ \cdots \circ \mathbf{K}^{(M)} + \beta^{-1}\mathbf{I}. \quad (14)$$

[1] For $M = 3$, and indexing elements of $\mathbf{w}$ by (l, m, n), (12) can be written as

$$y = \sum_{l,m,n} w_{l,m,n}\phi_l^{(1)}\phi_m^{(2)}\phi_n^{(3)} + \varepsilon,$$

where $\phi_j^{(i)}$ is an element of $\mathbf{\Phi}^{(i)}$, and is a function of $\mathbf{x}^{(i)}$.

Conditioned on the factors, the joint likelihood of a vector of outputs $\mathbf{y} = [y_1, ..., y_N]^T$ is Gaussian: $\mathbf{y}|\{\mathcal{X}_i\}_{i=1}^N \sim \mathcal{N}(0; \mathbf{K})$.

If all basis functions $\mathbf{\Phi}^{(i)}$ are linear, then the generative model is multilinear, and the GP represents a Bayesian form of multilinear regression. For general kernel functions, multifactor GP regression can be performed in the same manner as normal GP regression. Given training data $\mathcal{D}$, the predictive distribution for a new set of inputs in each of the factors, $\tilde{\mathcal{X}}$, is Gaussian, and are defined in terms of the kernel function as in (5) – (8). Generalizing the above discussion to non-zero mean functions is straightforward.

In the case where the inputs $\{\mathcal{X}_i\}_{i=1}^N$ are unknown and the outputs are high-dimensional, the model can be viewed as a GPLVM (Lawrence, 2005) with a structured latent space. As it is usually assumed that different subsets of the observations are represented by the same vector in certain latent factors.[2]

Since the product of valid kernel functions is also a valid kernel function (Stitson et al., 1998; Rasmussen & Williams, 2006), any valid kernels may be used for the individual factors. Although this is a known result, products of kernel functions are rarely used. The value of our formulation is that it leads to intuition as to how and why to multiply kernels, by considering the underlying generative model. Previous work provides guidance as to how to determine the generative model as well. For example, simple bilinear models have been used successfully to model stylistic variation in typefaces (Tenenbaum & Freeman, 2000), and multilinear models have also been used to capture the dependence of facial images on identity, lighting, and pose (Tenenbaum & Freeman, 2000; Vasilescu & Terzopoulos, 2002). Nonlinear manifolds are clearly useful for modeling the space of human poses (Elgammal & Lee, 2004; Grochow et al., 2004; Urtasun et al., 2005), but we may wish to express the dependence of motion data on other factors with linear kernels. In the next section, we use such experience with simpler models of motion capture data to guide the selection of kernel functions for more complex multifactor models.

3. A Model for Human Motion

In this paper, we apply the multifactor model to human motion capture data consisting of sequences of poses. A single pose is represented as a feature vector $\mathbf{y}_t$ of 89 dimensions, including 43 angular degrees-of-freedom (DOF) (see Figure 1), their velocities, and the global translational velocity. Joints with 3 DOFs and the global orientation are represented as exponential maps (Grassia, 1998); other joints are represented as Euler angles. An entire motion is represented as a sequence of T poses, $\mathbf{y}_{1:T}$.

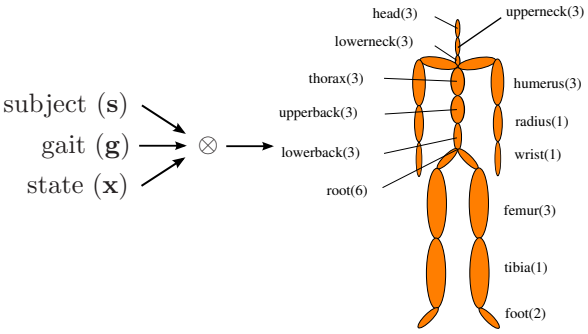

Figure 1. The skeleton used in our experiments is a simplified version of the default skeleton in the CMU mocap database. The numbers in parentheses indicate the number of DOFs for the joint directly above the labeled body node in the kinematic tree.

We focus on periodic human locomotion, such as walking and running, and model each pose in a motion sequence as arising from a combination of three independent factors:

- the identity of the subject performing the motion, represented as a 3D vector $\mathbf{s}$;
- the gait of locomotion (walk, stride, or run), represented as a 3D vector $\mathbf{g}$; and
- the current state in the motion sequence, represented as a 3D vector $\mathbf{x}$. For example, $\mathbf{x}$ corresponds to the phase of a cyclic gait.

For the purpose of the discussion, We will refer to $\mathbf{s}$ and $\mathbf{g}$ as the *style*, and $\mathbf{x}$ as the *content* of the motion. These latent input coordinates are not normally provided in observed motions. Hence, the model is a form of the GPLVM, in which we estimate the latent coordinates.

We must also choose the type of kernel functions for each set of input coordinates. Fortunately, we can draw on experience from previous work to help select the mappings. In particular, it has been shown that, for a style-specific model of motion, a nonlinear GPLVM model with an "RBF kernel" provides excellent results (Grochow et al., 2004; Urtasun et al., 2005), whereas linear models (such as obtained by linear PCA) do not capture the nonlinearities of human poses. Second, stylistic parameters can often be mod-

[2]For example, a set of distinct face images are assumed to share the same lighting direction, or a set of poses are assumed to share the same gait.

eled effectively using a linear space of styles (Brand & Hertzmann, 2000; Sidenbladh et al., 2000; Urtasun et al., 2006b) or multilinear in the case of multiple factors (Vasilescu, 2002). Third, since each DOF y_d may have a very different variance, it is important to introduce scale terms w_d for individual DOFs (Grochow et al., 2004). Based on these observations, we employ the following kernel function for the d-th DOF:

$$k_d([\mathbf{x},\mathbf{s},\mathbf{g}],[\mathbf{x}',\mathbf{s}',\mathbf{g}']) = \frac{1}{w_d^2}((\mathbf{s}^T\mathbf{s}')(\mathbf{g}^T\mathbf{g}')\exp(-\frac{\gamma}{2}||\mathbf{x}-\mathbf{x}'||^2)+\beta^{-1}\delta). \quad (15)$$

This defines a Gaussian process $f_d(\mathbf{x},\mathbf{s},\mathbf{g})$ for each pose DOF, which is assumed to be independent conditioned on the inputs. Note that, if we fix values of $\mathbf{s}$ and $\mathbf{g}$, we get a style-specific GP over poses $\mathbf{y}$ conditioned on the content $\mathbf{x}$.

For any particular motion sequence, we assume the style stays constant over time, and only model dynamics in the content space. We consider two approaches: nonlinear GP dynamics and a circle dynamics model (CDM), where the content vectors are restricted to lie on a unit circle (Elgammal & Lee, 2004). In the first approach, we assume the time-series obeys a nonlinear dynamical mapping:

$$\mathbf{x}_t = h(\mathbf{x}_{t-1}) + \varepsilon \quad (16)$$

Furthermore, we assume that h is a GP with a "linear + RBF" kernel; hence, for any given $\mathbf{s}$ and $\mathbf{g}$, the model is a GPDM (Wang et al., 2006).

In the CDM, low-dimensional coordinates are parameterized by a phase parameter θ_t, such that $\mathbf{x}_t = [\cos\theta_t, \sin\theta_t]^T$. Phase is linear as a function of time, parameterized by offset θ_0 and step-size $\Delta\theta$: $\theta_t = \theta_0 + t\Delta\theta$. Each sequence is then parameterized only by θ_0 and $\Delta\theta$. The step-size accounts for the different frequencies of different gaits. (The sampling rate of the motion capture data is the same in all cases).

Given training sequences, we learn the model by maximizing the log-posterior of the unknown factors $\mathbf{x}$, $\mathbf{s}$ and $\mathbf{g}$ for each pose, as well as the kernel parameters. As mentioned before, each motion sequence has a single $\mathbf{s}$ and a single $\mathbf{g}$ for each pose; these factors are not allowed to vary through time. Furthermore, motions performed by the same subject are constrained to have the same $\mathbf{s}$ as each other, and motions with the same type of gait are constrained to have the same $\mathbf{g}$. The β and γ hyperparameters have prior $p(\beta,\gamma) \propto (\beta\gamma)^{-1}$; all other hyperparameters and factors have uniform priors. Numerical optimization is performed using L-BFGS-B (Zhu et al., 1997). Note that we do not constrain corresponding poses in different sequences to share the same $\mathbf{x}$, as we do not assume prior knowledge of the exact correspondences. It is desirable, however, to restrict the content of different styles to lie on the same trajectory, especially for motion synthesis. This is the main motivation for the CDM.

Table 1. RMS errors for long prediction. Sequence indices correspond to the sequences in the CMU mocap database.

Model	GPDM		B-GPDM		CDM
Style	no	yes	no	yes	yes
07-02	1.56	0.91	1.75	0.76	0.38
08-04	1.18	0.48	1.30	0.97	0.47
08-05	1.91	0.56	1.29	0.57	1.77
08-11	2.42	1.06	1.52	1.36	0.80
07-04	1.10	1.10	1.17	1.32	0.72
07-12	1.45	1.06	1.39	0.78	0.57
37-01	1.04	0.75	0.98	0.91	0.35
16-35	1.41	0.53	0.55	0.40	0.39
09-07	1.34	0.49	0.87	0.67	0.57
Avg.	1.49	0.77	1.20	0.86	0.67

Table 2. RMS errors for short prediction (averaged over 24 samples).

Model	B-GPDM		CDM
Style	no	yes	yes
07-04	1.21 ± .035	0.92 ± .030	1.12 ± .048
07-12	1.48 ± .033	0.88 ± .037	1.14 ± .050
37-01	1.00 ± .026	0.70 ± .013	0.85 ± .019

4. Experiments

We now evaluate the ability of learned multifactor models to perform time-series prediction from motion capture data, and to synthesize new motion sequences in new styles.

4.1. Prediction

In the prediction task, we first learn models from a collection of motion clips.[3] Then, given a portion of a new sequence, we predict the subsequent frames of the sequence, and compare them against ground-truth. No timewarping is done on any of the training or testing

[3]The data are taken from CMU (mocaps.cs.cmu.edu) data sets 02_02, 02_03, 35_01, 35_18, 08_01, 08_07, downsampled by a factor of 4, and constitute 314 frames in total.

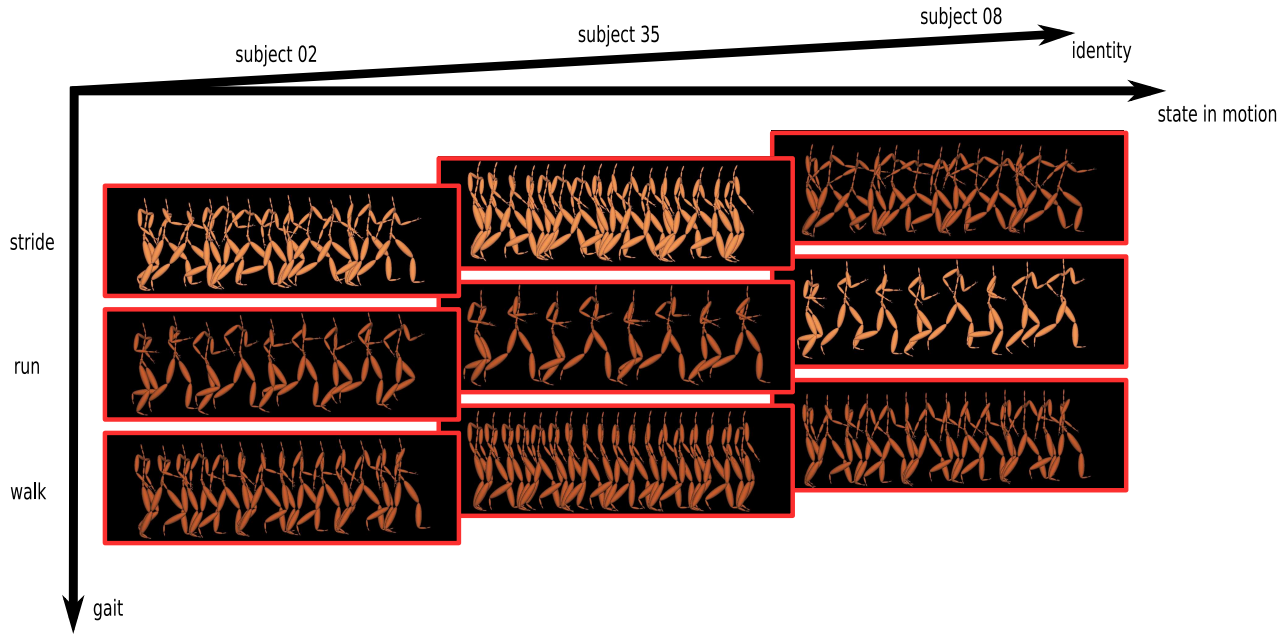

Figure 2. The structure of the multifactor model, where each sequence of poses are generated by an identity/subject vector, a gait vector, and a trajectory of states. Not all combinations of identity and gaits are available in the training data; the sequences (02, stride), (35, stride), and (08, run) are missing data inferred by the stylistic CDM.

data. We compare single-factor dynamical models for $\mathbf{x}_t$ which do not explicitly model style ($\mathbf{s}$ and $\mathbf{g}$) with the multifactor models introduced in the previous section (but use the same dynamical model in $\mathbf{x}_t$). We will refer to the latter as stylistic models here. The models compared include the GPDM, the B-GPDM (Urtasun et al., 2006a), and the CDM. The B-GPDM is a variant of GPDM which heavily prefers smooth trajectories in the latent space. Following previous work, we use 3D latent spaces for $\mathbf{x}_t$ in the GPDMs. The CDM restricts $\mathbf{x}_t$ to lie on a circle, and is therefore 2D. For the stylistic models, 2 additional 3D latent spaces are introduced, corresponding to the $\mathbf{s}$ and $\mathbf{g}$ factors.

Given that a 2D circle latent space is unable to model any stylistic variations, the CDM without style is omitted as it performed very poorly. Single-factor GPDMs with higher latent dimensionality are also omitted, as we have found that additional latent dimensions do not improve performance.

In prediction, the hidden factors are first estimated for the test subjects, by maximizing the joint posterior of all unknowns, conditioned on the test sequence and the learned model. Prediction is then performed by extrapolating the latent sequence of $\mathbf{x}_t$'s. For the GPDM variants, this is done by optimizing the joint dynamics distribution (Wang et al., 2006); for the CDM, this is done by taking the appropriate number of linear steps in phase. In both cases, the subject and gait are assumed to stay constant for the new sequence. New poses are then generated as the mean of the conditional Gaussian given the computed factors for each time-step.

All five models are tested by a long prediction experiment and a short prediction experiment. In long prediction, poses for just over half a cycle are provided, and poses for the next cycle are predicted.[4] The mean RMS errors of all of the predicted frames are shown in Table 1. The stylistic versions the GPDMs perform better than the single-factor versions by 48% and 28%, respectively. The stylistic CDM model achieved the lowest average rates among all models in the long prediction test.

In short prediction, only about a quarter of a cycle is given, and half a cycle is predicted.[5] Here we selected 3 data sets — all from subjects not seen in the training data — consisting of walks of varying speeds. For each set, 24 random starting poses are selected (constrained by the need for there to be enough ground

[4]For walking data, 25 frames are given, 40 are predicted. For running data, 13 are given, 20 are predicted. This is due to the difference in the number of poses per cycle.

[5]10 poses given, 20 poses predicted

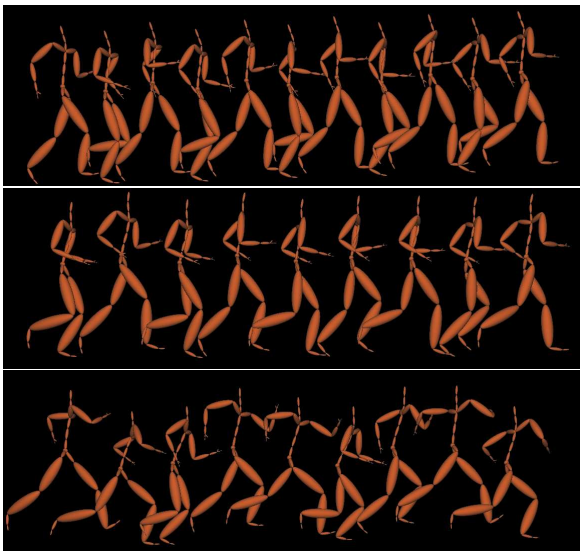

Figure 3. Motions generated from Gaussian sampling of the gait space and subject space. None of the poses are present in the training data.

truth data after the start pose), and we show the mean and standard error of the average RMS for each pose. Here we do not test the GPDM models, as they performed worse than the B-GPDM models in the previous test. The stylistic model improved upon the original B-GPDM model in all 3 data sets. The stylistic CDM model had a higher mean error, as well as more variability than the the stylistic B-GPDM model. This is in part due to its need for accurate estimation of the starting phase, as well as step size, which is less reliable for small numbers of input poses.

4.2. Motion Synthesis

The learned stylistic CDM can be used to generate motions not present in the training data. The model was learned from three subjects (02, 35, and 08 from the CMU database) all with some missing data. For subjects 02 and 35, the training data comprised examples of walking and running, but not striding. For subject 08, the training data included examples of walking and striding only. To generate new motion trajectories, a step size $\Delta\theta$ must be determined. We fit a bilinear model to the step sizes estimated during learning, mapping from $\mathbf{s}$ and $\mathbf{g}$ to $\Delta\theta$. Because the step-size determines the speed of the motion, we can generate motions of varying speeds.

Figure 2 depicts the structure of the multifactor model, including the inferred motions. These inferred motions are not simply copies of poses from a nearby gait or subject. In particular, the striding poses for subjects 02 and 35 contain stylistic elements of their respective walks: the inferred striding motion for 35 contains very little hand movement compared to the one striding training sequence (upper right), which is nevertheless consistent with subject 35's walking style (bottom center). Similarly, the bending of the left arm for subject 02 (upper left) is evident in that subject's walking style (lower left).

We can also generate new motions by random sampling. We fit one Gaussian distribution to the learned subject vector ($\mathbf{s}$) and another to the learned gait vector ($\mathbf{g}$), and then generate random new styles by sampling from these Gaussians. The step size $\Delta\theta$ can then be predicted by the bilinear model, and a sequence $\mathbf{x}_t$ of arbitrary length can then be generated.

The synthesized motions are shown in Figure 3. The top and middle rows are typical samples between a walk and a run. The bottom row is a slightly less typical, being a mixture of a run and a stride, which exaggerates the flight phase of running. In general, we find that convex combinations of styles produce reasonable motions.

Figure 4 demonstrates the ability of the model to generate smooth transitions from walking to running and from running to striding. The transitions are generated by linearly interpolating the gait vector with respect to the changing state vector. The subject vector is fixed to that of subject 02.

5. Discussion

We have described a multifactor regression and dimensionality reduction framework that unifies multilinear models with Bayesian regression and non-linear dimensionality reduction. The model can be viewed as a form of hierarchical Bayesian prior: modeling stylistic variation allows us to model a distribution of distributions, and thus generalize to new data with a style-specific distribution not included in the training data. In all of our experiments, we found that stylistic models performed better than generic models.

A number of potentially-daunting choices are involved in determining which factors and kernels to use. We have made these choices by considering subproblems and special cases of the underlying generative model, such as "style-only" models and "content-only" models. Considering these cases sheds light on how to combine these models, and we recommend this approach. Alternatively, model selection techniques could be employed, assuming a large dataset is available. Either way, we believe that multiplicative combination of kernels will be useful for modeling many types of data sources with multiple factors.

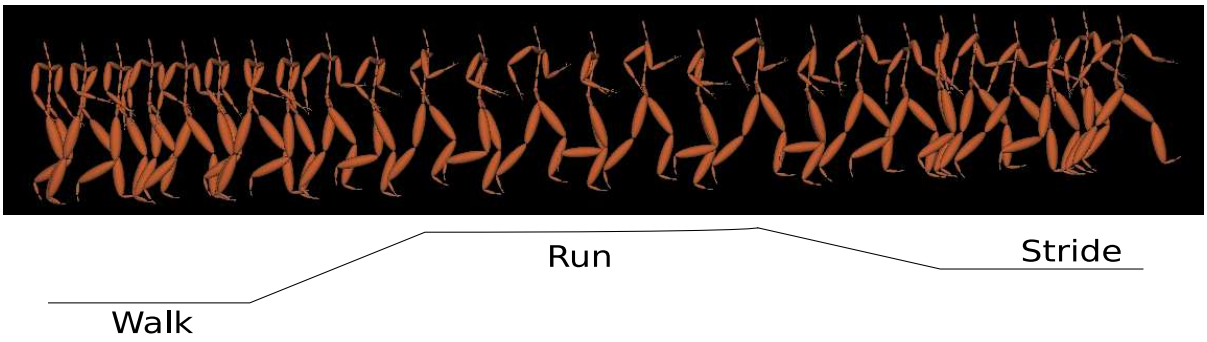

Figure 4. Transitions between different motions are achieved by linear interpolation in the gait space.

Acknowledgements

This project was funded in part by the Alfred P. Sloan Foundation, the Canadian Institute for Advanced Research, Canada Foundation for Innovation, Microsoft Research, NSERC Canada, and the Ontario Ministry of Research and Innovation. The data used in this project was obtained from mocap.cs.cmu.edu, which was created with funding from NSF EIA-0196217.

References

Brand, M., & Hertzmann, A. (2000). Style machines. *ACM SIGGRAPH* (pp. 183–192).

De Lathauwer, L., De Moor, B., & Vandewalle, J. (2000). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, *21*, 1253–1278.

Elgammal, A., & Lee, C.-S. (2004). Separating style and content on a nonlinear manifold. *IEEE Conf. Comp. Vis. and Pattern Recognition* (pp. 478–485). Vol. 1.

Grassia, F. S. (1998). Practical parameterization of rotations using the exponential map. *JGT*, *3*, 29–48.

Grochow, K., Martin, S. L., Hertzmann, A., & Popović, Z. (2004). Style-based inverse kinematics. *ACM Transactions on Graphics*, *23*, 522–531. Proc. SIGGRAPH.

Lawrence, N. D. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *J. Machine Learning Res.*, *6*, 1783–1816.

Li, Y., Du, Y., & Lin, X. (2005). Kernel-based multifactor analysis for image synthesis and recognition. *IEEE Inter. Conf. Comp. Vis. (ICCV)* (pp. 114–119). Vol. 1.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

Rose, C., Cohen, M. F., & Bodenheimer, B. (1998). Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, *18*, 32–40.

Rose, C., Sloan, P.-P. J., & Cohen, M. F. (2001). Artist-directed inverse-kinematics using radial basis function interpolation. *Computer Graphics Forum*, *20*. Proc. EG.

Shapiro, A., Cao, Y., & Faloutsos, P. (2006). Style components. *Graphics Interface* (pp. 33–40).

Sidenbladh, H., Black, M. J., & Fleet, D. J. (2000). Stochastic tracking of 3D human figures using 2D image motion. *ECCV* (pp. 702–718). Part II.

Stitson, M. O., Gammerman, A., Vapnik, V., Vovk, V., Watkins, C., & Weston, J. (1998). Support vector regression with ANOVA decomposition kernels. In B. Schölkopf, C. J. C. Burges and A. J. Smola (Eds.), *Advances in kernel methods: Support vector learning*. The MIT Press.

Tenenbaum, J. B., & Freeman, W. T. (2000). Separating style and content with bilinear models. *Neural Computation*, *12*, 1247–1283.

Urtasun, R., Fleet, D. J., & Fua, P. (2006a). 3D people tracking with Gaussian process dynamical models. *IEEE Conf. Comp. Vis. & Pattern Rec.* (pp. 238–245). Vol. 1.

Urtasun, R., Fleet, D. J., & Fua, P. (2006b). Temporal motion models for monocular and multiview 3D human body tracking. *CVIU*, *104*, 157–177.

Urtasun, R., Fleet, D. J., Hertzmann, A., & Fua, P. (2005). Priors for people tracking from small training sets. *IEEE Inter. Conf. Comp. Vis. (ICCV)* (pp. 403–410). Vol. 1.

Vasilescu, M. A. O. (2002). Human motion signatures: Analysis, synthesis, recognition. *Inter. Conf. Pattern Recognition (ICPR)* (pp. 456–460). Vol. III.

Vasilescu, M. A. O., & Terzopoulos, D. (2002). Multilinear analysis of image ensembles: TensorFaces. *Computer Vision – ECCV 2002* (pp. 447–460). Springer. Part I.

Vasilescu, M. A. O., & Terzopoulos, D. (2004). TensorTextures: Multilinear image-based rendering. *ACM Transactions on Graphics*, *23*, 336–342. Proc. SIGGRAPH.

Vlasic, D., Brand, M., Pfister, H., & Popović, J. (2005). Face transfer with multilinear models. *ACM Transactions on Graphics*, *24*, 426–433. Proc. SIGGRAPH.

Wang, J. M., Fleet, D. J., & Hertzmann, A. (2006). Gaussian process dynamical models. *Adv. Neural Information Processing Systems 18* (pp. 1441–1448). Proc. NIPS '05.

Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J. (1997). Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, *23*, 550–560.