Motion Models for People Tracking

David J. Fleet

Abstract This chapter provides an introduction to models of human pose and motion for use in 3D human pose tracking. We concentrate on probabilistic latent variable models of kinematics, most of which are learned from motion capture data, and on recent physics-based models. We briefly discuss important open problems and future research challenges.

1 Introduction

Prior information about human pose and motion has been essential for resolving ambiguities in video-based pose estimation and tracking. Motion estimation may be relatively straightforward if one is given several cameras and a constrained setting with minimal occlusion (*e.g.*, [8, 18, 30]), but the general monocular problem remains difficult without prior information. A prior model biases pose estimation toward plausible poses when pose might otherwise be under-constrained, or when measurements might be noisy, or missing due to occlusion. A good prior model should be sufficiently general to admit all (or most) plausible motions of the human body, but also strong enough to resolve ambiguities and alleviate the inherent challenges imposed by the high-dimensional estimation task. Finding the right balance between these competing goals is difficult. Most successful recent techniques for monocular pose tracking have focused on the use of strong, *activity-specific* prior models learned from human motion capture data.

This chapter provides a tutorial introduction to models of human pose and motion for video-based people tracking. We adopt a probabilistic framework, as it is perhaps the most straightforward and well-understood calculus for coping with uncertainty and fusing noisy sources of information. We first outline the basic probabilistic formulation, and then introduce the principal types of motion models.

Department of Computer Science
University of Toronto, Toronto, e-mail: fleet@cs.toronto.edu

1.1 Human Pose Tracking

From a single camera it is hard to escape depth-scale ambiguities, missing observations of body parts due to occlusion, and reflection ambiguities where different 3D poses produce similar images. Because of these sources of uncertainty, it has become common to formulate human pose tracking as a *Bayesian filtering* problem. As such, the goal is to approximate the posterior probability distribution over human poses or motions, given the image measurements (or observations).

Formally, let $\mathbf{x}_t$ denote the state of the body at time t. It represents the unknown parameters of the model we wish to estimate. In our case, the state typically comprises the joint angles of the body along with the position and orientation of the body in world coordinates. Different parametrizations of the joint angles are discussed in Chapter **??**, Section **??**. Tracking is formulated in terms of the posterior probability distribution over state sequences, $\mathbf{x}_{1:t} \equiv (\mathbf{x}_1, \ldots, \mathbf{x}_t)$, given the observation history, $\mathbf{z}_{1:t} \equiv (\mathbf{z}_1, \ldots, \mathbf{z}_t)$; *i.e.*,

$$p(\mathbf{x}_{1:t} \,|\, \mathbf{z}_{1:t}) \;=\; \frac{p(\mathbf{z}_{1:t} \,|\, \mathbf{x}_{1:t})\, p(\mathbf{x}_{1:t})}{p(\mathbf{z}_{1:t})} \,. \tag{1}$$

Here, the two key factors are $p(\mathbf{x}_{1:t})$, the prior motion model, and $p(\mathbf{z}_{1:t} \,|\, \mathbf{x}_{1:t})$, the likelihood model. The likelihood is the probability of observing the image measurements given a state sequence. In effect the likelihood provides a measure of the consistency between a hypothetical motion and the given image observations. The observations might simply be the image at time t, or they might be a collection of image measurements at time t (*e.g.*, image edge locations or optical flow). The denominator in (1), $p(\mathbf{z}_{1:t})$, does not depend on the state sequence, and can therefore be treated as an unknown constant for the purposes of this chapter.

Inference is the process of computing (or approximating) the posterior distribution (1), or estimating the most probable motion (*i.e.*, the *maximum a posteriori (MAP)* estimate). This is intractable for most pose tracking problems of interest. Even approximating $p(\mathbf{x}_{1:t} \,|\, \mathbf{z}_{1:t})$ is difficult because of the high dimensionality of the motion $\mathbf{x}_{1:t}$, and the observation sequence $\mathbf{z}_{1:t}$. For these reasons it is common to simplify the model, and therefore the computations required for inference.

One way to simplify inference is to assume that the observations are independent given the states. In other words, one assumes that the joint likelihood can be written as a product of simpler likelihoods, one for each time step:

$$p(\mathbf{z}_{1:t} \,|\, \mathbf{x}_{1:t}) \;=\; \prod_{i=1}^{t} p(\mathbf{z}_i \,|\, \mathbf{x}_i) \,. \tag{2}$$

For good generative models, which account for observations up to additive white noise, this is a reasonable assumption. But in many cases it is more a matter of convenience because it allows for more efficient inference, and the specification of the likelihood is typically more straightforward. Common measurement models and likelihood functions are discussed in Chapter **??**.

Given the conditional independence of the observations, we can express the posterior distribution at time t in terms of the likelihood at time t, the motion model, and the posterior at time $t-1$:

$$p(\mathbf{x}_{1:t} \,|\, \mathbf{z}_{1:t}) \;\propto\; p(\mathbf{z}_t \,|\, \mathbf{x}_t)\; p(\mathbf{x}_t \,|\, \mathbf{x}_{1:t-1})\; p(\mathbf{x}_{1:t-1} \,|\, \mathbf{z}_{1:t-1}) \;. \tag{3}$$

One can further simplify (3) by modeling motion as a first-order Markov process:

$$p(\mathbf{x}_t \,|\, \mathbf{x}_{1:t-1}) \;=\; p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}) \;. \tag{4}$$

While this is not strictly necessary, it greatly simplifies the formulation of motion models and inference process. In particular, it means that the posterior can be expressed recursively, where all past history of any significance is represented entirely within the posterior distribution at the previous time step.

Nevertheless, the number of unknowns in $\mathbf{x}_{1:t}$ grows linearly with the number of time steps, so for long sequences the posterior in (3) is difficult to compute. The size of the covariance matrix, for example, is quadratic in the dimension of $\mathbf{x}_{1:t}$. Another way to simplify inference is to focus solely on the state at the current time. This *marginal* posterior distribution, called the filtering distribution, is given by:

$$\begin{aligned} p(\mathbf{x}_t \,|\, \mathbf{z}_{1:t}) \;&=\; \int_{\mathbf{x}_{1:t-1}} p(\mathbf{x}_{1:t} \,|\, \mathbf{z}_{1:t}) \\ &\propto\; p(\mathbf{z}_t \,|\, \mathbf{x}_t) \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1} \,|\, \mathbf{z}_{1:t-1}) \;. \end{aligned} \tag{5}$$

Two main factors comprise the filtering distribution, namely, the likelihood, $p(\mathbf{z}_t \,|\, \mathbf{x}_t)$, and the prediction distribution, $p(\mathbf{x}_t \,|\, \mathbf{z}_{1:t-1})$, given by the integral in (5). The recursive form of the filtering distribution leads to well-known, online inference methods. The simplest such method, suitable for linear-Gaussian observation and motion models, is the well-known Kalman filter(*e.g.*, [43, 74, 80]). Unfortunately the Kalman Filter is not suitable for human pose tracking where the dynamics are usually nonlinear and likelihood functions are usually non-Gaussian and multi-modal.

A natural alternative for inference with non-Gaussian, multi-modal posterior distributions is the particle filter (*a.k.a.* sequential Monte Carlo methods [13, 19, 31]). Such methods approximate the filtering distribution with a weighted set of state samples, and then uses sample statistics to approximate expectation under the posterior or filtering distribution. They were first applied to visual tracking with the CONDENSATION algorithm [29]. They have since been used extensively for monocular tracking of 3D human pose with various likelihood functions and prior motion models (*e.g.*, [6, 11, 26, 27, 38, 40, 50, 57, 60, 61, 64]). For a more detailed discussion of sequential Monte Carlo methods, see the review article by Doucet *et al.*[13].

Finally, tracking typically requires a good initial guess for the pose in the first frame to initialize inference. Initial guesses are also useful to facilitate recovery from tracking failures. Methods for detecting people (see Chapter **??**), and discriminative methods for single-frame 3D pose estimation (see Chapter **??**) provide natural mechanisms to address these problems.

2 Kinematic Joint Limits and Smooth Motion

The kinematic structure of the human body permits a limited range of motion in each joint. Knees and elbows, for example, should not be hyper-extended under normal circumstances, and the torso cannot tilt or twist arbitrarily. One central role of a prior model is to ensure that the poses estimated from an image or image sequence will satisfy such biomechanical limits. While joint limits are often enforced using thresholds, imposed independently on each rotational DOF, the true nature of joint limits in the human body is more complex. In particular, joint limits are dynamic and dependent on the state of other joints [22]. Fortunately, depending on the joint parameterization, many joint constraints can be specified as linear inequalities. This is sometimes useful since, when combined with linear or quadratic objective criteria, one obtains a linear or quadratic programming problem (*e.g.*, see [10]).

While further research on joints limits is needed to understand general limits and individual variability, it appears clear that joint limits by themselves do not encode sufficient prior knowledge to facilitate tractable and robust inference of pose from monocular video (*e.g.*, [57]). Rather, we require some form of density model that captures the *plausibility* of feasible poses and motions under typical circumstances.

Perhaps the simplest prior model of human motion is a smooth, low-order Markov process (*e.g.*, [21, 48, 57, 74]). A common first-order model specifies that the pose $\mathbf{y}$ at time $t+1$ is equal to the pose at time t, up to additive Gaussian noise:

$$\mathbf{y}_{t+1} \;=\; \mathbf{y}_t + \eta \; . \tag{6}$$

The *process noise* η is usually assumed to be mean-zero, with covariance Λ, *i.e.*, $\eta \sim \mathcal{N}(0,\Lambda)$. It follows that the conditional density of $\mathbf{y}_{t+1}$ is $\mathbf{y}_{t+1} | \mathbf{y}_t \sim \mathcal{N}(\mathbf{y}_t, \Lambda)$. Equivalently, it follows that $p(\mathbf{y}_{t+1} | \mathbf{y}_t) = G(\mathbf{y}_{t+1}; \mathbf{y}_t, \Sigma)$ where $G(\mathbf{y}; \mu, \Lambda)$ is a Gaussian function, parameterized by its mean μ and covariance Λ, evaluated at $\mathbf{y}$. Second-order models exploit velocity for future predictions. That is, one can express $\mathbf{y}_{t+1}$ in terms of $\mathbf{y}_t$ and $\mathbf{y}_{t-1}$, often with a damping constant $0 < \kappa < 1$; *e.g.*,

$$\mathbf{y}_{t+1} \;=\; \mathbf{y}_t + \kappa(\mathbf{y}_t - \mathbf{y}_{t-1}) + \eta \; . \tag{7}$$

Damping helps control divergence when predictions occur over multiple time steps.

Equations (6) and (7) are linear models, the general form of which, *i.e.*,

$$\mathbf{y}_{t+1} = \sum_{\tau=1}^{L} \mathbf{A}_\tau \, \mathbf{y}_{t-\tau+1} + \eta \; , \tag{8}$$

is an L^{th}-order linear-Gaussian dynamical system (LDS). In most cases, the parameters of the transition model are set manually. For instance, one can set the matrices $\mathbf{A}_\tau$ to be diagonal, as in (6) and (7), and then assume a diagonal covariance matrix, Λ, that is fixed or increases in proportion to $||\mathbf{y}_t - \mathbf{y}_{t-1}||^2$ [11].

One can also learn dynamical models from motion capture data (*e.g.*, [44]). In this way one can capture the coupling between different joints. But LDS learning often suffers from over-fitting with high-dimensional state spaces. This is because the number of parameters in the transition matrices $\mathbf{A}_n$ is quadratic in the state dimension. Large data sets are usually necessary.

The main attraction with smooth LDS priors is their generality. They can be applied to a wide diversity of motions, which is useful when the activity is not known *a priori*. Nevertheless, LDS models are sometimes problematic since they are often too weak to adequately constrain people tracking. This is especially problematic with monocular videos where the image evidence is often weak. In constrained settings, where observations from three or more cameras are available, and occlusions are limited, such models have been shown to achieve satisfactory performance [11].

3 Linear Kinematic Models

When one knows or has inferred the type of motion being tracked (*e.g.*, see Chapter **??** on activity recognition), or the identity of the person performing the motion, one can apply prior models that are specific to the activity and/or the subject. The common approach is to learn models off-line (prior to tracking) from motion capture data. Typically one wants a low-dimensional *latent* parameterization of the pose, and a dynamical model that captures typical pose sequences (*i.e.*, motions).

To introduce the idea, consider a dataset $\mathscr{D} = \{\mathbf{y}^{(i)}\}_{i=1...N}$ comprising N poses $\mathbf{y}^{(i)} \in \mathbb{R}^D$, *e.g.*, from a motion capture acquisition system. Each training pose comprises the angles of each joint degree of freedom, and relevant aspects of global orientation and position with respect to the world coordinate frame.[1] Many activities of interest, like walking, exhibit strong regularities when repeated by the one or several people. As a result, one can posit that the data lie on or near some low-dimensional manifold in the (high-dimensional) pose space.

Principle Component Analysis (PCA) can be used to approximate poses in a low-dimensional subspace, using the sum of a mean pose, $\mu_{\mathscr{D}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{y}^{(i)}$, and a linear combination of basis vectors. For a data matrix $\mathbf{A}$, the i^{th} column of which is $\mathbf{y}^{(i)} - \mu_{\mathscr{D}}$, the singular value decomposition (SVD) factorizes $\mathbf{A}$ into orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$, with $\mathbf{U} \equiv [\mathbf{u}_1, ..., \mathbf{u}_D]$, and a diagonal matrix $\mathbf{S}$ containing singular values arranged in non-increasing order, such that $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$. Choosing the first B singular vectors $\{\mathbf{u}_j\}_{j=1...B}$ (*a.k.a.*, the *eigen-poses*), a pose is approximated by

$$\mathbf{y} \;\approx\; \mu_{\mathscr{D}} + \sum_{j=1}^{B} x_j \, \mathbf{u}_j \qquad (9)$$

where x_j are scalar coefficients and $B \ll D$ controls the fraction of the variance in $\mathbf{A}$ accounted for by the subspace approximation. As such, the estimation of the pose can be replaced by the estimation of the coefficients $\mathbf{x} \equiv [x_1, ..., x_B]$. Since B is

[1] Global position and orientation with respect to the world coordinate frame are somewhat arbitrary, and often excluded. Global orientation with respect to gravity, height above the ground, and the change in position with respect to the body-centric coordinate frame should be included.

typically much smaller than the dimensionality of the pose space D, pose estimation is greatly simplified.

In addition to the subspace pose model we need a dynamical model to capture the temporal evolution of pose. Perhaps the simplest such model is a LDS, like those in Section 2, but applied to the subspace parameters $\mathbf{c}$ rather than directly to the pose. The combination of a linear subspace projection (PCA) and a subspace LDS has been widely studied (*e.g.*, see [71]); in computer vision it is often referred to as a *Dynamic Texture* [12]. Most such models assume a first-order LDS, but higher-order models are sometimes useful [28]. The key advantage of the subspace dynamical model over the LDS model in (8) is the fact that the number of parameters in the transition matrices is quadratic in the dimension of the subspace rather than the dimension of the pose observations. Unfortunately, subspace LDS models do not capture nonlinearities that are common in many motions.

3.1 Motion PCA: Evolving Pose Subspace

Although modeling pose trajectories within a latent pose space can be difficult, modeling the motion directly is sometimes effective. That is, one can learn a linear, activity-specific kinematic model of the entire pose trajectory directly, rather than as a sequence of poses within a pose space. Originally formulated by Sidenbladh *et al.* [57], this approach has been used successfully in several ways [59, 66, 69].

As above, assume that each *pose* vector, $\mathbf{y} \in \mathbb{R}^D$, comprises joint angles and global DOFs. Writing the pose at time t as $\mathbf{y}_t$, we can express a *motion* as a vector comprising all joint angles throughout the entire sequence of M poses; *i.e.*,

$$\mathbf{m} \;=\; (\mathbf{y}_1^T, \cdots, \mathbf{y}_M^T)^T \; . \tag{10}$$

A training corpus typically involves multiple people performing the same activity multiple times. Because training motions occur at different speeds, or might be sampled at different rates, the first step of learning a model is to align and resample the training motions. For periodic sequences (*e.g.*, walking) one can use the fundamental frequency to determine the period (the duration of one cycle), and the phase needed for alignment. For non-periodic motions one can also manually segment and align the motions, or use some form of *dynamic time warping* (*e.g.*, see [46, 69]).[2] The canonical motion is then represented as a sequence of M (interpolated) poses, indexed by phase, $\phi_n \in (0,1]$, where $\phi_n = \frac{n}{M}$ and $0 \leq n < M$. Each training motion is a real-valued vector of length $D \times M$.

Given a collection of training motions, $\mathscr{D} = \{\mathbf{m}^{(i)}\}_{i=1}^N$, one can use PCA to form a subspace model. In this way a motion is expressed as a linear combination of a mean motion μ and a set of *eigen-motions*$\{\mathbf{b}_j\}_{j=1}^B$:

[2] Because the data are joint angles, interpolation is normally accomplished using *quaternion spherical interpolation* [56]. Naturally, the temporal sampling rate must be sufficiently high that one can interpolate the pose signal with reasonable accuracy.

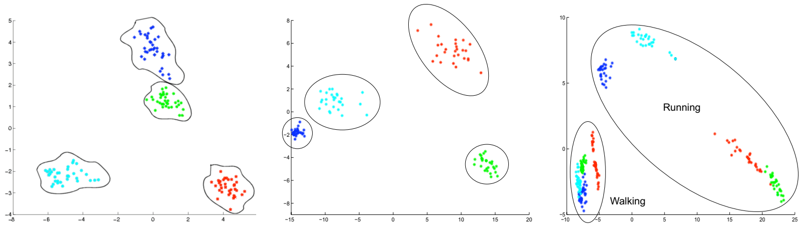

Fig. 1 Projections of training data onto the first two principal directions for a walk model (left), a run model (middle), and a model learned from walking and running data (right). Walking data comprised 4 mocap samples for each of 4 subjects (color coded) walking at 9 speeds varying from 3 to 7km/h. Running data were from the same subjects at 7 speeds ranging from 6 to 12km/h. Solid curves separating clusters are drawn for purposes of visualization. (Adapted from [69])

$$\mathbf{m} \;\approx\; \mu + \sum_{j=1}^{B} x_j \,\mathbf{b}_j \;\; . \tag{11}$$

The scalar coefficients, $\mathbf{x} = (x_1, ...x_B)^T$, characterize a particular motion. One typically chooses B so that a significant fraction (*e.g.*, 90%) of the empirical data variance is captured by the subspace projection. A pose is then defined to be a function of $\mathbf{x}$, the subspace coefficients, and the phase, ϕ; *i.e.*,

$$\mathbf{y}(\mathbf{c}, \phi) \;\approx\; \mu(\phi) + \sum_{j=1}^{B} x_j \,\mathbf{b}_j(\phi)\, . \tag{12}$$

Here, $\mathbf{b}_j$ denotes an eigen-motion, and $\mathbf{b}_j(\phi)$ is an *eigen-pose* at phase ϕ. Similarly, $\mu(\phi)$ is the mean pose at phase ϕ. In effect, the motion subspace yields a pose subspace that evolves as a function of ϕ. Nonlinearities in the evolution of the pose subspace are encoded implicitly in the eigen-motions.

With this model Sidenbladh *et al.* [57] formulated tracking as the estimation of global position, the speed of motion, the phase ϕ, and the subspace coefficients $\mathbf{x}$ at each frame. A particle filter was used for inference, to cope with transient pose ambiguities. Urtasun *et al.* [69] showed that motion-based PCA provides a convex model for many motions of interest such as walking and jogging (see Figure 1). That is, random draws from the underlying Gaussian model over the subspace coefficients produces plausible poses and motions. They also found that walks of different speeds for the same subject were tightly clustered in the subspace. This enabled motion-based recognition [69]. Troje [66] showed that this representation of walking facilitates the inference of other meaningful attributes, including gender and aspects of mental state. Sigal *et al.* [59] has since extended this to the inference of human attributes from video-based 3D pose data. But it is not clear how this representation can be extended to deal with different activities. Indeed, Urtasun *et al.* [69] showed that random samples drawn from a simple model learned from running and walking motions are not always plausible motions; *i.e.*, a Gaussian density function is not an adequate prior over multiple activities within a single subspace.

4 Nonlinear Kinematic Models

A periodic motion like walking follows a 1D cyclic trajectory in the high-dimensional pose space. Thus, while (linear) subspace models often require many dimensions to adequately span the empirical pose data, the underlying dimensionality of the motions may actually be significantly lower. One could, for example, parameterize position along a periodic pose trajectory with a 1D model. Allowing for variability, from cycle to cycle, or from person to person, one might posit that the poses lie on or near a low-dimensional, *nonlinear* manifold. The goal of a low-dimensional latent model is to parameterize the structure of the manifold. With nonlinear models one might be able to find more effective low-dimensional parameterizations than one might find with linear models.

The earliest nonlinear models used embedding methods for nonlinear dimensionality reduction (*e.g.*, [54]). Such methods provide low-dimensional latent positions for each training pose, but they do not provide a closed-form function that maps new poses to latent positions (often called out-of-sample extensions). Accordingly, methods based on nonlinear dimensionality reduction augment the embedding with learned mappings between the latent space and the observation (pose) space, along with a density model over the latent positions of the training poses (*e.g.*, [14, 60]). More recent methods, like the GPLVM, formulate and optimize a coherent model that incorporates the mappings, the embedding, and the density model.

4.1 Gaussian Process Latent Variable Model

The Gaussian Process Latent Variable Model (GPLVM)[3] is a nonlinear generalization of probabilistic PCA [33]. It is a generative latent variable model that comprises a low-dimensional latent space, and a stochastic, nonlinear mapping from the latent space to the original observation space. Conceptually, one hopes that the latent model captures the underlying *causes* for the high-dimensional training data. The GPLVM is useful for visualizing high-dimensional data [33], and it has been shown to generalize well even with small or moderate amounts of training data [68].

To explain the basic GPLVM it is easiest to first examine Gaussian Process (GP) regression[51]. To that end, consider a mapping from a vector-valued input, $\mathbf{x}$, to a scalar output, y. Let the mapping be expressed in parametric form as

$$y \;=\; g(\mathbf{x}) + \eta \;, \tag{13}$$

where η is mean-zero Gaussian, with variance β, and g has a generalized linear form. That is, let g be a weighted sum of nonlinear basis functions $\phi_j(\mathbf{x})$:

[3] http://staffwww.dcs.shef.ac.uk/people/N.Lawrence/gpsoftware.html is a comprehensive GPLVM code base. GPLVM code is also in the Matlab toolbox for dimensionality reduction available at http://homepage.tudelft.nl/19j49/

$$g(\mathbf{x}) \;=\; \sum_{j=1}^{J} w_j\,\phi_j(\mathbf{x}) \;=\; \mathbf{w}^T\boldsymbol{\Phi}(\mathbf{x}), \tag{14}$$

where $\mathbf{w} \equiv (w_1,\ldots,w_J)^T$, and the vector $\boldsymbol{\Phi}(\mathbf{x}) \equiv (\phi_1(\mathbf{x}),\ldots,\phi_J(\mathbf{x}))^T$ comprises the basis functions evaluated at $\mathbf{x}$. To complete the model, we assume a mean-zero Gaussian prior for $\mathbf{w}$ with unit covariance, $\mathbf{w} \sim \mathcal{N}(0;\mathbf{I})$, and we let the *noise* η be independent of $\mathbf{w}$.

Because y in Eq. 13 is a linear function of Gaussian random variables, it is also Gaussian, and therefore characterized by its mean and covariance. Because $\mathbf{w}$ and η are both mean-zero, it follows that y is mean-zero:

$$\mu(\mathbf{x}) \;\equiv\; E[y] \;=\; E[\mathbf{w}^T\boldsymbol{\Phi}(\mathbf{x})+\eta] \;=\; 0\,. \tag{15}$$

One can also show that, given two inputs, $\mathbf{x}$ and $\mathbf{x}'$, the covariance of their outputs, y and y', satisfies

$$\begin{aligned} k(\mathbf{x},\mathbf{x}') \;\equiv\; E[yy'] \;&=\; E[(\mathbf{w}^T\boldsymbol{\Phi}(\mathbf{x})+\eta)(\mathbf{w}^T\boldsymbol{\Phi}(\mathbf{x}')+\eta')] \\ &=\; \boldsymbol{\Phi}(\mathbf{x})^T\boldsymbol{\Phi}(\mathbf{x}') + \beta\,\delta(\mathbf{x},\mathbf{x}')\,, \end{aligned} \tag{16}$$

where δ is 1 when $\mathbf{x}$ and $\mathbf{x}'$ are the same inputs, and 0 otherwise. One can derive Eq. 16 using the model assumptions, $E[\mathbf{w}]=0$, $E[\mathbf{w}\mathbf{w}^T]=\mathbf{I}$, $E[w_j\eta]=0$, and $E[\eta^2]=\beta$. The functions $\mu(\mathbf{x})$ and $k(\mathbf{x},\mathbf{x}')$ are referred to as the mean function and the kernel (or covariance) function, respectively.

The mapping from $\mathbf{x}$ to y in (13) is a Gaussian Process (GP). It is a continuous stochastic process that is fully specified by its mean and covariance functions. For instance, with the appropriate choice of Gaussian basis functions [51], we obtain the well-known RBF kernel, combined with the variance of the additive noise:

$$k(\mathbf{x},\mathbf{x}') \;=\; \alpha\exp\left(-\frac{\gamma}{2}||\mathbf{x}-\mathbf{x}'||^2\right) + \beta\,\delta(\mathbf{x},\mathbf{x}')\,, \tag{17}$$

where the α, β and γ are the *hyperparameters* of the kernel; *i.e.*, α determines the magnitude of the covariance, γ determines the effective correlation length in the latent space, and β determines the variance of the additive noise. Alternative assumptions about the form of $\{\phi_j(\mathbf{x})\}$ in (14) lead to different kernel functions.

The GP model has several appealing properties. One stems from the formulation of $p(y\,|\,\mathbf{x})$ as the marginalization of $p(y,\mathbf{w}\,|\,\mathbf{x})$. By marginalizing over $\mathbf{w}$, *e.g.*, instead of estimating $\mathbf{w}$ using maximum likelihood, the GP mitigates over-fitting problems that commonly occur when one has only small or moderate amounts of training data. The GP also provides a measure of uncertainty in y (*i.e.*, the variance) which is useful in many applications. Finally, with the GP one does not have to specify the basis functions (*i.e.*, the features) directly. Rather, one only needs to specify the form of the kernel function [41, 51].

Suppose one is given IID training pairs, $\mathscr{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1\ldots N}$, with mean-zero outputs $y^{(i)}$. To learn a GP model one does not have to estimate $\mathbf{w}$, but one does have to estimate the kernel hyperparameters. This is usually done by maximizing the empirical data likelihood, *i.e.* the density over $\mathbf{z} \equiv (y^{(1)}, \dots, y^{(N)})^T$ conditioned on $\{\mathbf{x}^{(i)}\}$. It follows from the GP model that the data likelihood is mean-zero Gaussian with a covariance (kernel) matrix $\mathbf{K}$ having elements $\mathbf{K}_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$:

$$p(\mathbf{z} \,|\, \{\mathbf{x}^{(i)}\}, \theta) \;=\; \frac{1}{\sqrt{(2\pi)^N |\mathbf{K}|}} \exp\left(-\frac{1}{2} \mathbf{z}^T \mathbf{K}^{-1} \mathbf{z} \right) \,. \tag{18}$$

where θ is the vector of hyperparameters upon which $k(\cdot,\cdot)$ depends. Differentiating the log likelihood with respect to θ can be done in closed form, and hence can be used for optimization (*e.g.*, with *scaled conjugate gradient*).

For pose data, the GP outputs must be vector-valued, *i.e.*, $\mathbf{y}^{(i)} \in \mathbb{R}^D$. The training pairs are then given by $\mathscr{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1\dots N}$. If one uses the same kernel function for all output dimensions, then the joint likelihood function is the product of the likelihood for each output dimension. More specifically, let $\mathbf{Y} = [\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(N)}]^T$, and let $\mathbf{y}_d$ be the d^{th} column of $\mathbf{Y}$; *i.e.*, $\mathbf{y}_d$ comprises the d^{th} element of each of the N training outputs. Then, one can write the joint GP likelihood as the product of likelihoods for each dimension $\mathbf{y}_d$:

$$p(\mathbf{Y} \,|\, \{\mathbf{x}^{(i)}\}, \theta) \;=\; \prod_{d=1}^{D} \frac{1}{\sqrt{(2\pi)^N |\mathbf{K}|}} \exp\left(-\frac{1}{2} \mathbf{y}_d^T \mathbf{K}^{-1} \mathbf{y}_d \right) \,, \tag{19}$$

where θ is the vector of kernel hyperparameters. By using the same kernel matrix for each observation dimension we greatly reduce the number of hyperparameters. Further, a common kernel naturally captures correlations among the different output dimensions that depend directly on the corresponding latent positions. That is, although the conditional distribution is the product of 1D densities, the different observation dimensions are not independent. Rather, they depend on the common kernel matrix. That said, when modeling pose data, different dimensions (*e.g.*, joint angles) have significantly different variances. In this case, it is useful to discard the common scale parameter (α in (17)), and instead use a separate scale parameter for each observation dimension (*e.g.*, see [20, 68]).

GP regression is a *supervised* model, where training data include both $\mathbf{x}$ and $\mathbf{y}$. The GPLVM is an *unsupervised* model, where $\mathbf{Y}$ is the only available training data [33]. Learning a GPLVM therefore entails the estimation of a latent representative (position) for each training sample, in addition to the hyperparameters θ. Lawrence [33] showed that for linear features, *i.e.*, $\Phi(\mathbf{x}) = \mathbf{x}$, the GPLVM is equivalent to probabilistic PCA. In this sense the GPLVM is a generalization of probabilistic PCA to nonlinear mappings.

GPLVM learning entails numerical optimization to maximize the joint posterior $p(\mathbf{Y} \,|\, \{\mathbf{x}^{(i)}\}, \theta)\, p(\{\mathbf{x}^{(i)}\})\, p(\theta)$ with respect to $\{\mathbf{x}^{(i)}\}$ and θ. The prior over the latent representatives is typically a broad Gaussian density. The prior over the hyperparameters is typically uninformative, unless domain-specific knowledge is available. An initial guess for the optimization is often critical; one can use PCA or some other form of nonlinear dimensionality reduction method like LLE [54]. Usually the dimension of the latent space is chosen manually.

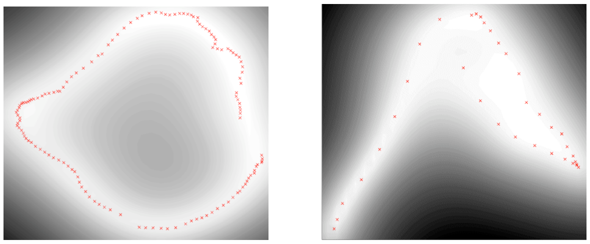

Fig. 2 GPLVM latent spaces learned from mocap data: (left) one walk cycle and (right) a golf swing. Red crosses are optimized latent points $\mathbf{x}^{(i)} \in \mathbb{R}^2$. Grayscale depicts $-D \ln \sigma^2(\mathbf{x}) - \mathbf{x}^T\mathbf{x}$; lighter points imply a lower variance (22) and hence more likely poses. (Adapted from [68])

A key property of the GPLVM is its predictive distribution. Given a new latent position, $\mathbf{x}$, the distribution over the observation space is Gaussian, with a simple closed-form expression for its mean and covariance:

$$\begin{aligned} \mathbf{y}\,|\,\mathbf{x},\mathbf{Y},\{\mathbf{x}^{(i)}\} \;&\sim \mathcal{N}(\mathbf{m}(\mathbf{x});\sigma^2(\mathbf{x}))\,, && (20)\\ \text{where} \qquad \mathbf{m}(\mathbf{x}) \;&=\; \mathbf{Y}^T\;\mathbf{K}^{-1}\mathbf{k}(\mathbf{x})\,, && (21)\\ \sigma^2(\mathbf{x}) \;&=\; k(\mathbf{x},\mathbf{x}) - \mathbf{k}(\mathbf{x})^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}), && (22)\\ \mathbf{k}(\mathbf{x}) \;&=\; (\,k(\mathbf{x},\mathbf{x}^{(1)}),\,...,\,k(\mathbf{x},\mathbf{x}^{(N)})\,)^T\,. && (23) \end{aligned}$$

The predictive distribution is central to inferring a new pose. Effectively, these equations show that, given a latent position $\mathbf{x}$, the mean prediction for $\mathbf{y}$ in Eq. 21 is just a weighted sum of training poses; the weights are a function of the kernel distances between $\mathbf{x}$ and the latent training representatives, along with the pre-computed, inverse kernel matrix $\mathbf{K}^{-1}$. One can also use this predictive distribution to find the latent position $\mathbf{x}$ that is maximally consistent with a given pose $\mathbf{y}$.[4]

Another useful expression is the likelihood of a new pair $(\mathbf{x},\mathbf{y})$, since during tracking we often require the estimation of both quantities. In particular, up to an additive constant, the negative log probability of a pair $(\mathbf{x},\mathbf{y})$, given $\mathbf{Y}$ and $\{\mathbf{x}^{(i)}\}$, is

$$L(\mathbf{x},\mathbf{y}) \;=\; \frac{\|(\mathbf{y}-\mathbf{m}(\mathbf{x}))\|^2}{2\sigma^2(\mathbf{x})} + \frac{D}{2}\ln\sigma^2(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}\|^2\,. \qquad (24)$$

Minimizing $L(\mathbf{x},\mathbf{y})$ therefore aims to minimize reconstruction errors (*i.e.*, to keep $\mathbf{y}$ close to the mean $\mathbf{m}(\mathbf{x})$), while keeping latent positions close to the training data (*i.e.*, to keep $\sigma^2(\mathbf{x})$ small). The third term in (24) is the prior over latent positions that usually has relatively little influence on the optimized latent positions. Figure 2 depicts this log likelihood for GPLVMs learned from a walk and a golf swing.

For visual tracking one can combine a suitable log likelihood term for the image data, with the log prior over new points, $L(\mathbf{x},\mathbf{y})$, in order to formulate an objective

[4] The GPLVM has a closed-form mapping from $\mathbf{x}$ to $\mathbf{y}$, but there is no closed-form inverse mapping. As a consequence, optimization is required to find the optimal $\mathbf{x}$ for a given $\mathbf{y}$.

function. Because $L(\mathbf{x},\mathbf{y})$ is easily differentiated one can use continuous optimization to find MAP estimates [68], or one can use a sequential Monte Carlomethod for inference [50]. During tracking one usually estimates both $\mathbf{x}$ and $\mathbf{y}$ at each frame. In some cases one may wish to search only over $\mathbf{x}$, using the deterministic mapping from $\mathbf{x}$ to $\mathbf{y}$ given by the mean, $\mathbf{m}(\mathbf{x})$ [67]. This has the advantage that one searches a much smaller state space, but it comes with the disadvantage that one is explicitly limited to a linear combination of the training poses with no additional stylistic variability.

4.2 Gaussian Process Dynamical Model

The GPLVM is formulated for IID training data, drawn fairly from the true pose density over the observation space. By ignoring the obvious temporal coherence that is significant in human motion, the GPLVM often produces models in which consecutive poses do not always correspond to nearby points in the latent space. Conversely, one might expect a good model to map smooth pose trajectories to smooth latent trajectories, thereby facilitating temporal prediction and effective tracking. The Gaussian Process Dynamical Model (GPDM)[5], as the name suggests, is an extension of the GPLVM to incorporate temporal structure for times-series data, thereby promoting smoothness in the latent representation of motion.

The GPDM replaces the IID prior over inputs $\{\mathbf{x}^{(i)}\}$ with a Gaussian Process prior over latent trajectories. For example, let latent positions at time t be predicted by a first-order model, defined by a matrix $\mathbf{A}$, a feature vector $\Psi(\mathbf{x})$, and Gaussian noise, $\eta \sim \mathcal{N}(\mathbf{0},\xi\mathbf{I})$:

$$\mathbf{x}_t \;=\; \mathbf{A}\,\Psi(\mathbf{x}_{t-1})+\eta\ . \tag{25}$$

For linear features, $\Psi(\mathbf{x}_t)=\mathbf{x}_t$, this model (25) reduces to an auto-regressive model(*c.f.*, Eq. 8). But like the GPLVM, one can incorporate nonlinear features and analytically marginalize out the weights $\mathbf{A}$ (assuming a Gaussian prior over the columns of $\mathbf{A}$). This provides a GP prior over the latent sequences that correspond to training motions. (See [67, 77] for the mathematical details.)

The GPDM combines a nonlinear mapping from latent points to observations, with nonlinear dynamical predictions. Marginalizing over the weight matrices of both mappings helps reduce potential over-fitting problems. Learning entails the estimation of a latent position for each training pose, with the hyperparameters for the latent mapping and the dynamical model. Figure 3 depicts a GPDM learned from three gait cycles of walking. Color in Figure 3 (middle) is analogous to the greylevel in Figure 2. Warmer colors (red) indicate small variances, hence more likely poses. Cooler colors (blue) indicate larger variances and hence unlikely poses. Like the GPLVM, GPDM predictions are analytical and straightforwardly combined with an image likelihood for pose tracking [67]. Figure 4 depicts the monocular estimation of walking, despite significant occlusion by the bushes on the left side of the image.

[5] GPDM Code: `http://www.dgp.toronto.edu/~jmwang/gpdm/`

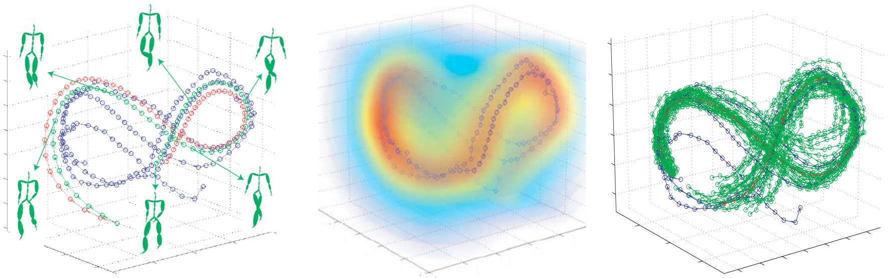

Fig. 3 A 3D GPDM is learned from 3 walk cycles. (Left) The latent positions of training poses are shown as blue circles. (Middle) The pose variance as a function of latent position is color coded, with red (blue) points having small (large) variance. (Right) Each green trajectory is a random sample from the latent dynamical model; the mean motion of which is the red trajectory in the left plot. (Adapted from [77])

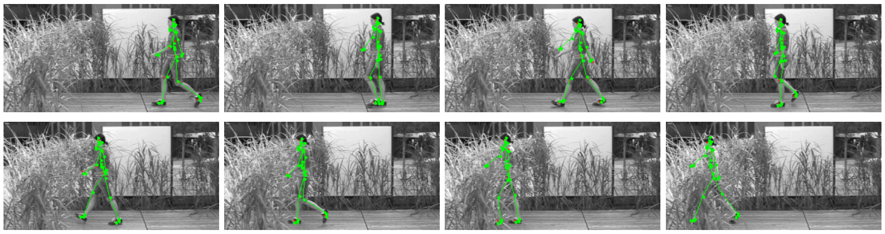

Fig. 4 Monocular tracking results with a GPDM learned from walking data. The 3D person is tracked despite the almost total occlusion by the bush on the left side of the image, where only the head is visible by the end of the sequence. (Adapted from [67]).

4.3 Constrained Latent Spaces and Other Variants

The GPLVM does not work well with large datasets because learning and inference are, respectively, cubic and quadratic in the number of training poses. Approximations to the covariance matrix can be used to improve efficiency (*e.g.*, [49]), but their use requires care, since local minima often fail to produce useful models. Similar approximations to the GPDM have not been formulated.

A second issue concerns the sensitivity of the GPLVM and GPDM optimizations to the initial guess, and the fact that many local minima do not represent useful models [77] (*e.g.*, see Figure 5 (left)). Such local minima are especially problematic when there is significant stylistic variability in the training data. Given the number of unknowns in the learning problem, and the lack of structure imposed on the latent representation, this problem is not particularly surprising.

To address these issues, several interesting GPLVM variants have appeared in recent years. They demonstrate some of the ways in which one can impose more structure on the latent representation in order to produce more useful models.

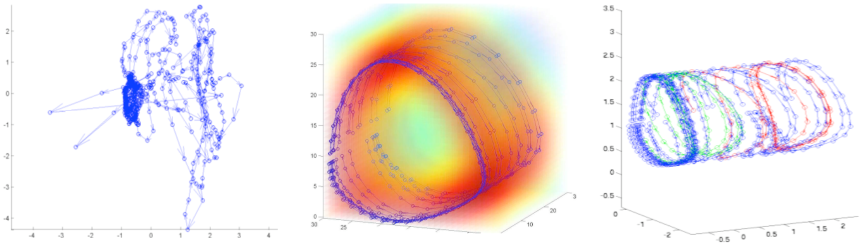

Fig. 5 (Left) A GPDM is learned from mocap data of people walking and running. The latent trajectories are not smooth, and trajectories drawn from the dynamical model are not realistic motions. (Middle) This GPDM is constrained to lie on a cylindrical topology, with an LLE prior that encourages nearby poses to remain close in the latent space. (Right) Random trajectories simulated by the model (in red and green) produce plausible motions. (Adapted from [70])

4.3.1 Back-Constraints and Topological Constraints

The GPLVM ensures that nearby latent positions map to similar poses. The converse is not true; similar poses do not necessarily get mapped to nearby latent positions. Despite the use of a dynamical prior, even the GPDM does not always produce smooth models with useful temporal predictions. To ensure smooth latent models, Lawrence and Quiñonero-Candela [35] introduced *back-constraints*. They suggested that one might parameterize latent position in terms of a smooth function of the observation space. For example, one might write the j^{th} coordinate of the i^{th} latent position as $x_{ij} = h_j(\mathbf{y}^{(i)}; \mathbf{a}_j)$, where $\{\mathbf{a}_j\}_{j=1...d}$ denotes the parameters of the mapping, and d is the dimension of the latent space. For instance, h might be expressed as a form of kernel-based regression, so nearby poses in the observation space map to similar latent positions. Rather than directly estimating the latent positions, learning a back-constrained GPLVM entails the estimation of the mapping parameters $\{\mathbf{a}_j\}$, by maximizing the empirical data likelihood.

Back-constraints can be used to model temporal dependence, thereby ensuring that time-series data will be mapped to smooth latent trajectories. They can also be used to specify latent topological structure. For instance, Urtasun *et al.* [70] used back-constraints to parameterize a cylindrical latent topology when modeling cyclic gaits, like walking and running. They also incorporated local, *soft back-constraints* to encourage nearby poses to map to nearby latent positions. This is done in much the same way that LLE optimizes low-dimensional positions to maintain distances to nearby points in the observation space.

The combination of the cylindrical topology and the preservation of local neighborhoods produces the latent representation depicted in Figure 5 (middle). This model captures running and walking performed by multiple subjects. Random samples from the model appear natural, including transitions between walking and running (*e.g.*, Figure 5 (right)). By comparison, the GPDM has difficulty coping with such stylistic diversity; GPDMs like that in Figure 5 (left) are typical for these training data, and do not produce plausible gaits.

4.3.2 Multi-Factor GPLVM

One way to capture significant stylistic diversity is to blend models that capture individual styles. For example, motivated by linear style interpolation and multilinear models (*e.g.*, [53, 65, 72]), one might consider a weighted sum of GPs, $\{g_i(\mathbf{x})\}$:

$$y = \sum_i s_i \, g_i(\mathbf{x}) + \eta = \sum_i s_i \, \mathbf{w}_i^T \Phi(\mathbf{x}) + \eta = \sum_i \sum_j s_i \, w_{ij}^T \phi_j(\mathbf{x}) + \eta \, . \tag{26}$$

This is a generative model for y with latent variables $\mathbf{z} = (\mathbf{x}, \mathbf{s})$. The latent space is composed of two subspaces, one for the blending weights $\mathbf{s} = (\ldots, s_i, \ldots)$, representing style, and one for $\mathbf{x}$ which captures the phase dependence of the pose.

If we assume Gaussian weight vectors, $\mathbf{w}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and Gaussian process noise, $\eta \sim \mathcal{N}(0, \beta)$, then it follows that $\mathbf{y}$ is a mean-zero GP with covariance function

$$k(\mathbf{z}, \mathbf{z}') = \mathbf{s}^T \mathbf{s}' \, \Phi(\mathbf{x})^T \Phi(\mathbf{x}') + \beta \delta(\mathbf{z}, \mathbf{z}') \, . \tag{27}$$

where δ is 1 when $\mathbf{z}$ and $\mathbf{z}'$ correspond to the same measurements, and 0 otherwise. The covariance function in Eq. 27 has two key factors, namely, the linear kernel on $\mathbf{s}$, and the nonlinear kernel on $\mathbf{x}$. This two-factor, scalar GP model is readily generalized to three or more factors, and to vector-valued outputs [76]. Each factor is associated with an individual latent subspace, and the covariance function (27) involves the product of one kernel for each factor.

Such multi-factor GPLVMs are particularly useful for mocap data where *side information* is often available. That is, for each mocap sample one typically knows the type of gait (*e.g.* run, walk, jog), as well as the subject's identity, age, weight, *etc.*, all of which contribute to the motion style. In a multi-factor GPLVM, each type of side information would be represented as a separate latent factor. As an example, Wang *et al.* [76] learned a three-factor model, using the subject's *identity*, the *gait* type (walk, stride or jog), and the *phase* of the gait cycle. All motions of one individual, independent of gait and phase, are constrained to share the same latent position in the identity subspace. All walking motions, independent of the subject and phase, share the same position in the gait subspace. And so on. With side information used in this way, the multi-factor GPLVM imposes structure on the latent space; structure that the GPDM would be unlikely to discover. As a result, multi-factor models tends to converge more readily to useful kinematic models, for different datasets and initial conditions.

Interestingly, one can view the multi-factor GPLVM as a Bayesian generalization of multilinear models (*e.g.*, [65, 72]). The two models are very similar when one uses linear features (*e.g.*, $\Phi(\mathbf{x}) = \mathbf{x}$ in (26)). The keys differences are that the GPLVM marginalizes over the weights (*i.e.*, the multilinear core tensor), which reduces the number of the unknowns that must be estimated and mitigates potential over-fitting problems. The second difference is that the multi-factor GPLVM generalizes naturally to nonlinear features (*c.f.*, [15]). When designed properly it is also possible to express the kernel matrix as product of much smaller kernels [75], greatly reducing the complexity of learning and inference.

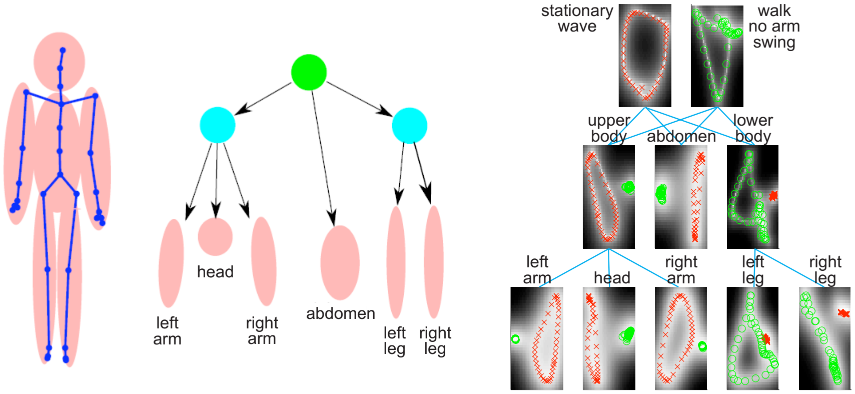

Fig. 6 In a hierarchical GPLVM, a latent position at one node provides Gaussian densities over its descendants. Here it is used to coordinate different body parts, for two activities, waving while standing still, and walking no arm swing. Red and green points, respectively, depict the latent positions at each node that correspond to poses from these two activities. (Adapted from [9])

4.3.3 Hierarchical GPLVM

Lawrence and Moore [34] proposed a hierarchy of GPLVMs in which latent positions at one level are specified by the output of a GP at the next level. This is another way to impose structure on a latent representation. One use of the Hierarchical GPLVM (hGPLVM) is to capture temporal coherence [34]. An initial GP maps time, or the gait phase, to a Gaussian density over positions in a latent pose space. A second GP then maps position in the latent pose space to a Gaussian density over pose in the original observation space. A temporal model like this has been used successfully for tracking in [1].

The hGPLVM could be used to model coordination between interacting people. The pose (or motion) of each person might be modeled by two separate GPLVMs. To coordinate their motions, a third GP simultaneously specifies Gaussian densities over the latent positions in the two person-specific GPLVM latent spaces.

One could also use the hGPLVM to model the coordination of different parts of the body, like that depicted in Figure 6 [34]. This model has six GPLVMs at the lowest level of the hierarchy (leaves of the tree). Each is responsible for one part of the body, mapping a latent position to a Gaussian density over pose (of its corresponding part). At the next level there are latent models that specify the coordination of the legs and of the upper body. The *lower body* model outputs Gaussian densities over latent positions within the left and right leg models to control leg swing. The hierarchy also includes multiple activities. In Figure 6 (right) the two activities are waving while the legs are standing still, and walking with no appreciable arm swing. Notice that the intermediate nodes of the hierarchy capture the latent structure of body parts for both activities. This hierarchical model of human motion was used successfully in [9] for tracking a person walking while waving an arm, thereby composing a new motion from elements of the two training motions.

4.4 Switching Linear Dynamical Systems (SLDS)

One way to model data in the vicinity of a smooth, low-dimensional manifold, is to use local linear models, much as one might approximate a smooth curve with a piecewise linear function. One such model for time-series data, like human motion, is the *switching linear dynamical system* (*e.g.* [16, 44, 45, 47]). A Switching Linear Dynamical Systems (SLDS) comprises a set of LDS models and a discrete switching variable that specifies which LDS is active at each time. Each LDS captures the evolution of pose within a local region of the pose space, and can be viewed as an atomic *moveme*. During tracking one maintains a probability distribution over the switching variable, and a Gaussian density over pose for each LDS. If one marginalizes out the switching variable, one obtains a Gaussian mixture model over pose.

SLDS models are attractive for their intuitive simplicity, but they require large datasets and can be hard to learn. For each LDS (see Eq. 8) one requires a transition matrix and a covariance matrix for the process noise. For a D-dimensional pose vector there are $O(D^2)$ parameters for each transition matrix and for each covariance matrix. An SLDS with N components also requires $O(N^2)$ parameters to specify the temporal transition matrix for the switching variable. Hence, the number of unknowns to be optimized is large. One also faces a difficult model selection problem as one needs to decide how many LDS components to use in the model. Over-fitting and local minima are significant problems when learning SLDS models.

Li *et al.* advocate a model that addresses some of these shortcomings [36, 37, 38]. First, they express each linear component as a (latent) subspace LDS. Each components has a low-dimensional subspace that is learned with factor analysis (or PCA), and a LDS that models the evolution of the subspace coordinates. Second, the different local subspace models are configured to form a consistent global model using the Coordinated Mixture of Factor Analyzers[6] [55, 73]. Learning is formulated using variational Bayes, which also enables the automatic determination of the number of linear components and their dimensions. Li *et al.* demonstrated the effectiveness of this model for monocular tracking in [37, 38].

One interesting property of this class of models is its potential to model diverse styles and activities. For example, Figure 7 depicts a model learned from a 2405 frame training sequence of 56D human mocap data. The sequences comprising 5 activities, namely jumping jacks, twisting, bending, squats and jogging. Figure 7 (bottom) depicts the activity labels throughout the sequence. The learning algorithm automatically selected 11 subspace-LDS components, each with 7 dimensions. Figure 7 (top) depicts the most likely assignment of pose to each of the 11 components. Notice how the 11 components decompose the data into coherent atomic motions, each of which appears to be specific to a single activity. The last segment was captured by a single component since the jogging was done in place with minimal limb movement [36]. Such multi-activity models have not yet been used for video-based tracking with complex motion sequences.

[6] Code for the coordinated mixture of factor analyzers is included in the Matlab toolbox for dimensionality reduction available at `http://homepage.tudelft.nl/19j49/`

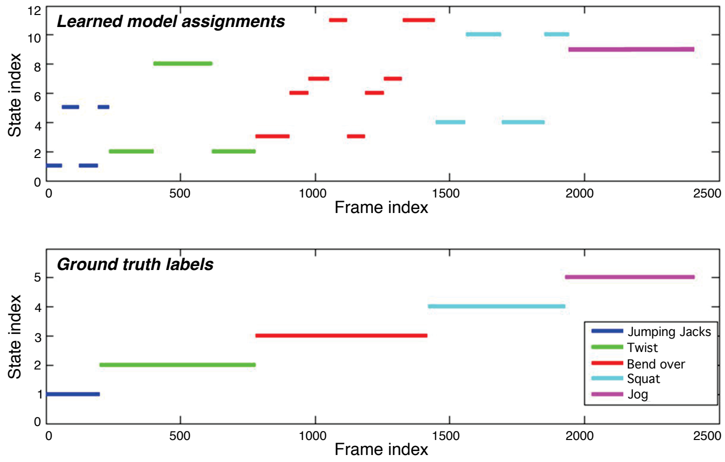

Fig. 7 (Bottom) Ground truth activity labels for a 2405 frame mocap sequence comprising five distinct activities. (Top) The most probable state of the switching variable for an 11-component SLDS model learned using Li's variational Bayes formulation. (From Li [36])

4.5 Conditional Restricted Bolzmann Machines

A third promising class of latent variable models has recently emerged based on the Restricted Bolzmann Machine (RBM) (*e.g.*, [23, 25]). An RBM is a probabilistic graphical model. It comprises a bipartite graph over the observation (visible) variables and the latent variables. As a result, conditioned on the state of the latent variables, the observation variables are independent of one another, and *vice versa*. In the usual RBM all variables are binary-valued, but it can be extended to real-valued observations, and is therefore applicable to modeling human pose. With its bipartite structure, RBM learning and inference are efficient. Learning is linear in the number of training exemplars with an algorithm known as *contrastive divergence* [23].

The Conditional Restricted Bolzmann Machines (CRBM) is an extension of the RBM to model time-series data[7] [63]. This is accomplished by conditioning the latent and observation variables at time t on the observations at the previous N time steps (for an N^{th}-order model). The *implicit mixture of CRBMs* (imCRBM) [62, 64] is an extension of the CRBM to include latent style variables. These style variables, much like those in the multi-factor GPLVM, modulate the weights (interaction potentials) of the CRBM in order to achieve distinct motion styles. If one marginalizes over these style variables one obtains a mixture of CRBMs (*i.e.*, an imCRBM).

Like the coordinated mixture of factor analyzers above, imCRBM learning can be supervised or unsupervised. When supervised, the style or activity labels are provided. In the unsupervised case the model discovers atomic motion primitives from the training data. An impressive diversity of styles can be learned and used for synthesis [62]. A variation of the model was used for monocular tracking in [64].

[7] Code: `http://www.cs.nyu.edu/~gwtaylor/code/`

Figure 8 depicts the behavior of a CRBM and an imCRBM in combination with a basic particle filter for monocular pose tracking. The input video (HUMANEVA S3, combo [58]) begins with walking and then transitions to jogging around frame 400. All models were trained on walking and jogging data from the same subject (S3), but with no transitions. Figure 8 (top-left) depicts RMSE for 3D joint position as at each frame for four trackers: 1) an annealed particle filter for baseline comparison; 2) a plain CRMB; 3) a supervised imCRBM (*i.e.*, imCRBM-2L) trained *with* walk and jog activity labels; and 4) an unsupervised imCRBM with 10 latent activity labels (*i.e.*, imCRBM-10U). CRBM-based models perform better than baseline. The two imCRBM with activity-specific components are more reliable than the basic CRBM in tracking the motion through the transition from walking to running.

Figure 8 (top-right) depicts the approximate posterior distribution over activity labels for the supervised model (imCRBM-2U) and the unsupervised model (imCRBM-10U). Uncertainty is evident in the vicinity of the walk-jog transition. Also notice that the unsupervised model appears to have discovered activity labels that correspond to coherent atomic movements. Interestingly they appear to be specific to the activity and the phase of the gait cycle. The bottom rows of Figure 8 depict MAP estimates of a particle filter with the imCRBM-2U motion prior.

While learning is challenging with sophisticated models like the CRBM and the imCRBM [24], this is an interesting class of models. Like the SLDS, RBM models are parametric, and thus do not suffer from having to store all the training data (as does the non-parametric GPLVM for instance). Furthermore, inference is very fast, and learning is linear in the number of training samples. As a consequence the CRBM and imCRBM can be trained on very large mocap corpora.

4.6 Heterogeneity, Compositionality, and Exogenous Factors

Most state-of-the-art approaches to tracking human motion rely on learned kinematic models. This is true of generative models and of discriminative model techniques (see Chapter **??**). With the development of new models and learning algorithms, recent methods for people tracking have produced very encouraging results. Nevertheless, important issues remain. Existing models only work well with a handful of activities, and modest stylistic diversity. They remain unable to model human motion over extended sequences in which people seamlessly transition from activity to activity. Generalization to a wide range of motion styles is similarly lacking.

The lack of *compositionality* in current models is one of the key barriers to improved generalization. For example, because limbs move with some degree of independence, there are myriad ways one might compose leg and arm movements. People usually walk with a counter-phase oscillation of the arms and legs. Sometimes they walk with relatively little arm swing, *e.g.*, if carrying a heavy object. And sometimes they walk with a hand raised, waving to a friend. A compositional model would model the elementary parts of the body, along with the ways they might be composed to form the whole. This would avoid the combinatorial explosion in the

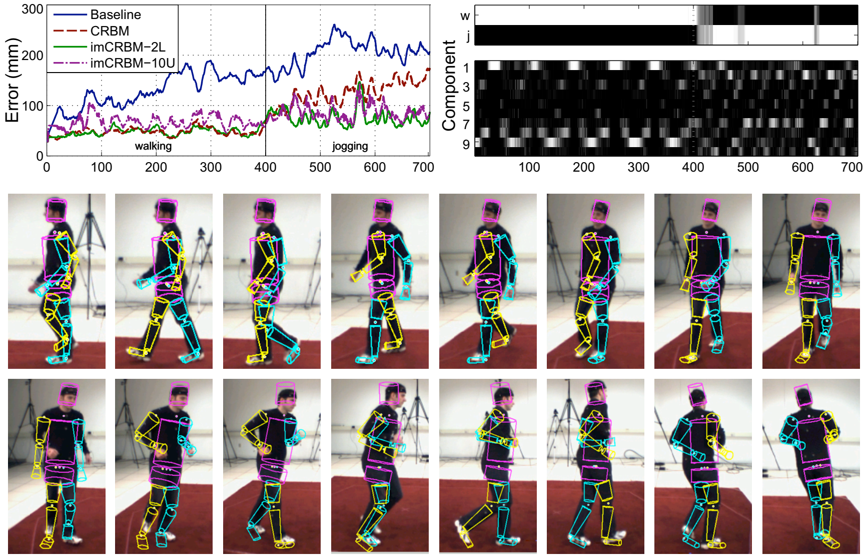

Fig. 8 (top-left) RMSE for monocular pose tracking based on CRBM and imCRBM for a HumanEva sequence with walking and jogging. (top-right) Posterior distribution over activity labels for a supervised imCRBM with two labels (walk/jog), and for an unsupervised imCRBM with 10 latent activities. The unsupervised model learns coherent motion primitives. (bottom 2 rows) Output of the particle filter with the supervised imCRBM prior motion model. (Adapted from [64])

size of training datasets that one would otherwise have to collect to model human motion holistically. Other than the hierarchical GPLVM model, all of the models described above are holistic, not compositional (*c.f.*, [9]).

Another issue concerns generalization with respect to exogenous factors. Not surprisingly, human motion is often more variable in natural environments than in the laboratory. People do not walk the same way on a slippery ice rink as they would on the underlying concrete pad once the ice is removed. They lean while carrying heavy objects or walking up a steep hill. The motion of one person may also depend greatly on other nearby people, or on external objects like the ball that one attempts to drive with the swing of a baseball bat. Current kinematic motion models do not generalize naturally when such factors are in play. They do not maintain balance, adapt to ground slope or surface roughness for example. As a consequence, the 3D motions estimated with kinematic models are sometimes overtly implausible. Visible artifacts in tracking walking people include jerky motions, pose estimates for which the feet float above or penetrate the ground plane, foot-skate (where feet slide along the ground), and out-of-plane rotations that violate balance.

One way to build richer kinematic models is to gather much more mocap data, *e.g.*, with varying ground slope, compliance, friction, roughness, loads or scene constraints, *etc.* But it remains unclear whether one would be able to collect such a voluminous amount of training data. And if one could do so, it is unclear how learning algorithms would be able to cope with the shear size of the resulting training corpus.

5 Newtonian (Physics-Based) Models

One way to mitigate some of the shortcomings of kinematic models is to incorporate constraints on motion and multi-body interactions based on Newtonian principles. For an articulated body with pose $\mathbf{y}$, the *equations of motion* from classical mechanics comprise a system of ordinary differential equations that relate accelerations, denoted $\ddot{\mathbf{y}}$, to forces:

$$\mathbf{M}\ddot{\mathbf{y}} \;=\; \mathbf{f}_{joints} + \mathbf{f}_{gravity} + \mathbf{f}_{contact} + \mathbf{a}\ . \tag{28}$$

The mass matrix $\mathbf{M}$ depends on the mass, inertial properties and pose of the body. The right side of Eq. 28 includes internal joint forces (or torques), due mainly muscle activations. The external forces acting on the body include forces due to gravity $\mathbf{f}_{gravity}$ and external contact. Contact forces in turn depend on surface geometry and the dynamics of the contact interface between two bodies, like stiffness and friction for example. Finally, $\mathbf{a}$ denotes generalized coriolis and centrifugal forces that occur with rotation and angular momentum. The equations of motion are somewhat tedious to derive properly, but articulated bodies typically permit textbook formulations. Importantly, many of these forces can be derived from first principles, and they provide important constraints on motion and interactions.

When combined with a suitable control mechanism, physics-based models offer several advantages over kinematic models. First, physics-based models should ensure that estimated motions are physically plausible, mitigating problems associated with foot placement and balance for example. Second, physics-based models should generalize in ways that are difficult for purely kinematic models. For example the change in body orientation that occurs as one carries a heavy object or walks down a steep hill should occur naturally to maintain balance. Third, the use of Newtonian and biomechanical principles of human locomotion may greatly reduce the current reliance on large corpora of human motion capture data. Indeed, many important characteristics of human locomotion can be attributed to optimality principles that produce stable, efficient gaits (*e.g.*, [7, 32]). Last, but not least, interactions and environmental factors are central to physics-based models, so one should be able to exploit such models to simultaneously infer both the motion and the properties of the world with which the subject interacts.

Despite their potential, there is relatively little work on physics-based models for people tracking.[8] One barrier stems from the complexity of full-body dynamics and contact (*e.g.*, [5]). Sensitivity to initial conditions, integration noise, and motion discontinuities at collisions mean that full-body simulation and control entail significant computational challenges. This remains true of modern humanoid robotics, biomechanics (*e.g.*, [52]), and character animation (*e.g.*, [39]).

[8] Several papers have used elastic solid models with depth inputs and a Kalman filter (*e.g.*, [43, 80]); but these domains involve relatively simple dynamics with smooth, contact-free motions.

5.1 Planar Models of Locomotion

Fortunately, there are reasons to believe that there exist low-dimensional abstractions of human locomotion which might be suitable for people tracking. Research in biomechanics and robotics has shown that the dynamics of bipedal walking is well described by relatively simple, planar, *passive-dynamic walking* models. Early models, such as those introduced by McGeer [42], were entirely passive and could walk downhill solely under the force of gravity; stable, bipedal walking is the natural limit cycle of their dynamics. Powered extensions of such models have since been been built and studied to explore the biomechanical principles of human locomotion [7, 17, 32, 42]. They walk stably on level-ground, exhibiting human-like gaits and energy-efficiency, and they can be used to model the preferred relationship between speed and step-length in human walking [32].

Inspired by these abstractions, Brubaker *et al.* [2, 3] developed two models of human walking, the *Anthropomorphic Walker* and the *Kneed Walker* (see Figure 9). These models exhibit essential physical properties, namely balance and ground contact, while walking and running with human-like gaits and efficiency. The Kneed Walker comprises a torso, two legs with knees, and a rounded foot that rolls along the ground to simulate the effects of ankle articulation. The model's kinematic, mass and inertial parameters are drawn from the biomechanics literature [42, 52]. Model forces are parameterized as linear torsional springs (*i.e.*, joint-based PD controllers).

One fascinating property of such models is that a good prior model can be found through controller optimization, rather than fitting mocap data. Brubaker *et al.* [2] were able to optimize many controllers for different operating points (*e.g.*, ground slopes, locomotion speeds and step-lengths), thereby defining an effective manifold of control settings. Their probabilistic model was defined in the vicinity of this manifold by adding Gaussian noise to the optimal control parameters. A random gait could then be produced by randomly drawing the control parameters and simulating the model using the equations of motion. This dynamics model is low-dimensional and exhibits stable human-like gaits with realistic ground contact, all in the 2D sagittal plane. The 3D kinematic model is then constrained to be consistent with the planar dynamics, and to move smoothly in its remaining degrees of freedom (DOF).

Tracking was performed using physical simulation within a particle filter (see Figure 9). The tracker handled occlusion, varying gait styles, and turning, producing realistic 3D reconstructions. With lower-body occlusions, it still produced realistic reconstructions and estimate the time and location of ground contact. When applied to the benchmark HUMANEVA dataset, monocular tracking errors in joint position are in the 65mm-100mm range [3]. Importantly, the prior model for this tracker does not rely on mocap training data from the same subjects performing the same motions like most other techniques that have been tested on HUMANEVA.

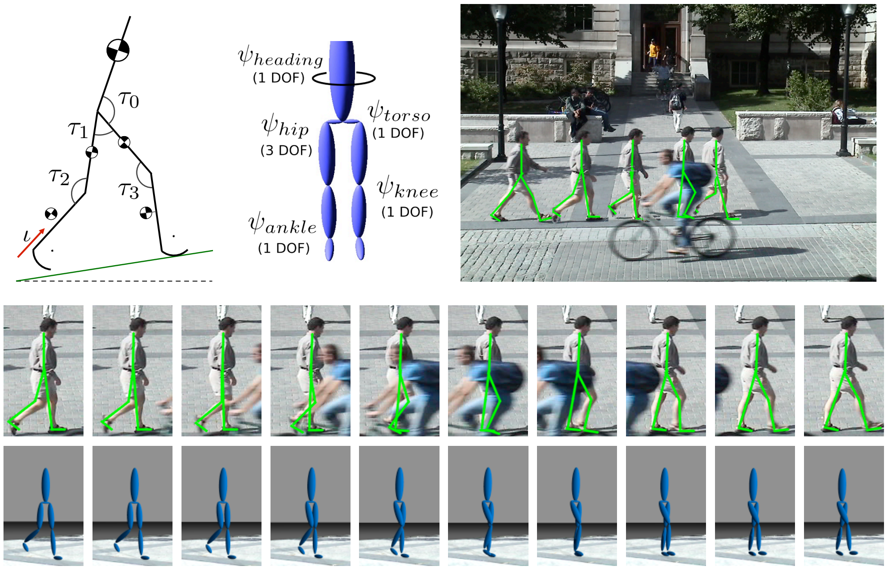

Fig. 9 (Top row) Composite of image sequence showing a walking subject and an occluding cyclist. The green stick figure in the right composite depicts on the MAP estimate of the pose on selected frames. (Bottom two rows) Cropped frames around occlusion. The green skeleton and blue 3D rendering depict the recovered MAP trajectory. (Adapted from [2])

5.2 Discussion: 3D Full-Body Models

Recent research has begun to consider physics-based models for full-body 3D control, motivated in part by the success of optimal planar models [2, 3] and the state-space SIMBICON controller [81]. In particular, Wang *et al.* [78, 79] have shown that human-like bipedal motion can be obtained by optimizing joint-space controllers with a collection of objective criteria motivated by empirical findings in biomechanics. The resulting motions appear reasonably natural, and adapt readily to different body morphologies (*e.g.*, tall or short), different environmental constraints (*e.g.*, walking on ice, or a narrow beam), and to various forms of uncertainty in either environmental conditions (*e.g.*, wind or surface roughness) or internal noise (*e.g.*, neural motor noise). While fascinating, such controllers are difficult to learn with over a hundred degrees of freedom, and they have not yet exhibited the degree of stylistic variation that one might need to track arbitrary people.

Another largely untapped research direction concerns the inference of human interactions with the environment. Brubaker *et al.* [4] have recently proposed a generic framework for estimating external forces due to gravity and surface contact from human motion. They define a generic measure of physical realism for human motion, and optimize various exogenous factors (*e.g.*, gravity, ground plane position and orientation) that are necessary to maximize realism. Initial results on motion capture data are very good, and results on video-based motion information are encouraging.

With general 3D formulations like this we might hope to build models of human motion that readily cope with ambiguity and noise without resorting to activity-specific latent variable models that are commonly used today.

6 Discussion

Progress in modeling human motion has been significant over the last decade, but many research directions remain unexplored. As discussed above, kinematic models have to move beyond activity-specific motions to much more complex sequences of multiple activities and natural transitions between them. Compositionality is largely unexplored, as is the related issue concerning a suitable computational definition of atomic motion primitives, in terms of which complex motions can be decomposed.

The use of dynamics is in its infancy. Open questions include the use of full-body 3D control mechanisms, and the ability to use physical principles to help detect and infer human interactions. Finding good control mechanisms appears essential for modeling human motion with effective, low-dimensional parameterizations. Physics-based models should also apply biological motion in general, since the basic principles of locomotion appear common to bipeds and quadrapeds. Finally, there are many potential ways in which physics might be augmented with kinematic properties learned from motion capture data.

Acknowledgements Much of the material in this chapter draws on work and perspectives that are the consequence of many fruitful research collaborations on visual tracking and modeling human motion. Thanks to Michael Black, Marcus Brubaker, Aaron Hertzmann, Geoff Hinton, Allan Jepson, Hedvig Kjellström. Neil Lawrence, Roland Memisevic, Leonid Sigal, Graham Taylor, Niko Troje, Raquel Urtasun, and Jack Wang. I also gratefully acknowledge financial support from NSERC Canada and the Canadian Institute for Advanced Research (CIfAR).

References

1. M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *Computer Vision and Pattern Recognition*, 2010.
2. M.A. Brubaker and D.J. Fleet. The kneed walker for human pose tracking. In *Computer Vision and Pattern Recognition*, 2008.
3. M.A. Brubaker, D.J. Fleet, and A. Hertzmann. Physics-based person tracking using the anthropomorphic walker. *International Journal of Computer Vision*, 87(1-2):140–155, 2010.
4. M.A. Brubaker, L. Sigal, and D.J. Fleet. Estimating contact dynamics. In *International Conference on Computer Vision*, pages 2389–2396, 2009.
5. M.A. Brubaker, L. Sigal, and D.J. Fleet. Physics-based human motion modeling for people tracking: A short tutotial. *Notes from IEEE ICCV Tutorial*, 2009. (Available from http://www.cs.toronto.edu/~ls/iccv2009tutorial/).
6. K. Choo and D.J. Fleet. People tracking using hybrid Monte Carlo filtering. In *International Conference on Computer Vision*, volume II, pages 321–328, 2001.
7. S.H. Collins and A. Ruina. A Bipedal Walking Robot with Efficient and Human-Like Gait. In *International Conference on Robotics and Automation*, 2005.

8. S Corazza, L Muendermann, A Chaudhari, T Demattio, C Cobelli, and T Andriacchi. A markerless motion capture system to study musculoskeletal biomechanics: visual hull and simulated annealing approach. *Annals of Biomedical Engineering*, 34(6):1019–1029, 2006.
9. J. Darby, B. Li, N. Costens, D.J. Fleet, and N.D. Lawrence. Backing off: Hierarchical decomposition of activity for 3d novel pose recovery. In *British Machine Vision Conference*, 2009.
10. M. de La Gorce, D.J. Fleet, and N. Paragios. Model-based 3d hand pose estimation from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page (to appear), 2011.
11. J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205, 2005.
12. G. Doretto, A. Chiuso, Y.N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.
13. A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.
14. A.M. Elgammal and C.-S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *Computer Vision and Pattern Recognition*, volume 2, pages 681–688, 2004.
15. A.M. Elgammal and C.-S. Lee. Separating style and content on a nonlinear manifold. In *Computer Vision and Pattern Recognition*, volume 1, pages 478–485, 2004.
16. E.B. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky. Nonparametric bayesian learning of switching linear dynamical systems. In *Advances in Neural Information Processing Systems*, pages 457–464, 2008.
17. R.J. Full and D.E. Koditschek. Templates and Anchors: Neuromechanical Hypotheses of Legged Locomotion on Land. *Journal of Experimental Biology*, 202:3325–3332, 1999.
18. J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Optimization and filtering for human motion capture. *International Journal of Computer Vision*, 87(1-2):75–92, 2010.
19. N.J. Gordon, D. J. Salmond, and A F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F: Radar and Signal Processing*, 140:107–113, 1993.
20. K. Grochow, S.L. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. *ACM Transactions on Graphics*, 23(3):522–531, 2004.
21. S. Hauberg, S. Sommer, and K.S. Pedersen. Gaussian-like spatial priors for articulated tracking. In *European Conference on Computer Vision*, volume 1, pages 425–437, 2010.
22. L. Herda, R. Urtasun, and P. Fua. Hierarchical implicit surface joint limits for human body tracking. *Computer Vision and Image Understanding*, 99(2):189–209, 2005.
23. G.E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.
24. G.E. Hinton. A practical guide to training restricted boltzmann machines. Technical Report UTML TR 2010-003, Department of Computer Science, University of Toronto, 2010.
25. G.E. Hinton, S. Osindero, and Y.W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.
26. S. Hou, A. Galata, F. Caillette, N.A. Thacker, and P.A. Bromiley. Real-time body tracking using a gaussian process latent variable model. In *International Conference on Computer Vision*, 2007.
27. N.R. Howe, M.E. Leventon, and W.T. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In *Advances in Neural Information Processing Systems*, pages 820–826, 1999.
28. M. Hyndman, A.D. Jepson, and D.J. Fleet. Higher-order autoregressive models for dynamic textures. In *British Machine Vision Conference*, 2007.
29. M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
30. L. Kakadiaris and D. Metaxas. Model-based estimation of 3D human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1453–1459, 2000.
31. A. Kong, J. S. Liu, and W. H. Wong. Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.

32. A.D. Kuo. A Simple Model of Bipedal Walking Predicts the Preferred Speed–Step Length Relationship. *Journal of Biomechanical Engineering*, 123(3):264–269, 2001.
33. N.D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.
34. N.D. Lawrence and A.J. Moore. Hierarchical Gaussian process latent variable models. In *International Conference on Machine Learning*, pages 481–488, 2007.
35. N.D. Lawrence and J. Quiñonero-Candela. Local distance preservation in the gp-lvm through back constraints. In *International Conference on Machine Learning*, pages 513–520, 2006.
36. R. Li. *Simulataneous learning of non-linear manifold and dynamical models for high-dimensional time series*. PhD thesis, Boston University, 2009.
37. R. Li, T.-P. Tian, S. Sclaroff, and M.-H. Yang. 3d human motion tracking with a coordinated mixture of factor analyzers. *International Journal of Computer Vision*, 87(1-2):170–190, 2010.
38. R. Li, M.-H. Yang, S. Sclaroff, and T.-P. Tian. Monocular tracking of 3d human motion with a coordinated mixture of factor analyzers. In *European Conference on Computer Vision*, volume 2, pages 137–150, 2006.
39. C.K. Liu, A. Hertzmann, and Z. Popović. Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics*, 24(3):1071–1081, 2005.
40. Z. Lu, M.A. Carreira-Perpin, and C. Sminchisescu. People tracking with the laplacian eigenmaps latent variable model. In *Advances in Neural Information Processing Systems*, pages 1705–1712, 2007.
41. D.J.C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.
42. T. McGeer. Dynamics and Control of Bipedal Locomotion. *Journal of Theoretical Biology*, 163:277–314, 1993.
43. D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.
44. B. North, A. Blake, M. Isard, and J. Rittscher. Learning and classification of complex dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1016–1034, 2000.
45. S.M. Oh, J.M. Rehg, T.R. Balch, and F. Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *International Journal of Computer Vision*, 77(1-3):103–124, 2008.
46. W. Pan and L. Torresani. Unsupervised hierarchical modeling of locomotion styles. In *International Conference on Machine Learning*, page 99, 2009.
47. V. Pavlovic, J.M. Rehg, and J. MacCormick. Learning switching linear models of human motion. In *Advances in Neural Information Processing Systems*, pages 981–987, 2000.
48. E. Poon and D.J. Fleet. Hybrid Monte Carlo filtering: edge-based people tracking. In *Workshop on Motion and Video Computing*, pages 151–158, 2002.
49. J. Quiñonero-Candela and C.E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.
50. L.M. Raskin, E. Rivlin, and M. Rudzsky. Using Gaussian process annealing particle filter for 3d human tracking. *EURASIP J. Adv. Sig. Proc.*, 2008.
51. C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. 2006.
52. D.G.E. Robertson, G.E. Caldwell, J. Hamill, G. Kamen, and S.N. Whittlesey. *Research Methods in Biomechanics*. Human Kinetics, 2004.
53. C. Rose, M.F. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18(5):32–40, 1998.
54. S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(550):2323–2326, 2000.
55. S.T. Roweis, L.K. Saul, and G.E. Hinton. Global coordination of local linear models. In *Advances in Neural Information Processing Systems*, pages 889–896, 2001.
56. K. Shoemake. Animating Rotation with Quaternion Curves. In *ACM Transactions on Graphics*, pages 245–254, 1985.

57. H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision*, volume 2, pages 702–718, 2000.
58. L. Sigal, A.O. Balan, and M.J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1-2):4–27, 2010.
59. L. Sigal, D.J. Fleet, N. Troje, and M. Livne. Human attributes from 3d pose tracking. In *European Conference on Computer Vision*, 2010.
60. C. Sminchisescu and A. Jepson. Generative modeling for continuous non-linearly embedded visual inference. In *International Conference on Machine Learning*, pages 759–766, 2004.
61. C. Sminchisescu and B. Triggs. Kinematic Jump Processes for Monocular 3D Human Tracking. In *Computer Vision and Pattern Recognition*, volume 1, pages 69–76, Madison, 2003.
62. G.W. Taylor and G.E. Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In *International Conference on Machine Learning*, 2009.
63. G.W. Taylor, G.E. Hinton, and S.T. Roweis. Modeling human motion using binary latent variables. In *Advances in Neural Information Processing Systems*, pages 1345–1352, 2006.
64. G.W. Taylor, L. Sigal, D.J. Fleet, and G.E. Hinton. Dynamical binary latent variable models for 3d human pose tracking. In *Computer Vision and Pattern Recognition*, pages 631–638, 2010.
65. J.B. Tenenbaum and W.T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.
66. N. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision*, 2(5):371–387, 2002.
67. R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models. In *Computer Vision and Pattern Recognition*, volume 1, pages 238–245, 2006.
68. R. Urtasun, D. J. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. In *International Conference on Computer Vision*, volume 1, pages 403–410, 2005.
69. R. Urtasun, D.J. Fleet, and P. Fua. Motion models for 3D people tracking. *Computer Vision and Image Understanding*, 104(2-3):157–177, 2006.
70. R. Urtasun, D.J. Fleet, A. Geiger, J. Popovic, T. Darrell, and N.D. Lawrence. Topologically-constrained latent variable models. In *International Conference on Machine Learning*, pages 1080–1087, 2008.
71. P. Van Overschee and B. De Moor. N4SID : Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.
72. M.A.O. Vasilescu. Human motion signatures: Analysis, synthesis, recognition. In *International Conference on Pattern Recognition*, volume III, pages 456–460, 2002.
73. J.J. Verbeek. Learning nonlinear image manifolds by global alignment of local linear models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1236–1250, 2006.
74. S. Wachter and H. H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3):174–192, 1999.
75. J.M. Wang. *Locomotion synthesis methods for humanoid characters*. PhD thesis, University of Toronto, 2010.
76. J.M. Wang, D.J. Fleet, and A. Hertzmann. Multifactor Gaussian process models for style-content separation. In *International Conference on Machine Learning*, pages 975–982, 2007.
77. J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008.
78. J.M. Wang, D.J. Fleet, and A. Hertzmann. Optimizing walking controllers. *ACM Transactions on Graphics*, 28(5), 2009.
79. J.M. Wang, D.J. Fleet, and A. Hertzmann. Optimizing walking controllers for uncertain inputs and environments. *ACM Transactions on Graphics*, 29(4), 2010.
80. C.R. Wren and A. Pentland. Dynamic models of human motion. In *International Conference on Automatic Face and Gesture Recognition*, pages 22–27, 1998.
81. K. Yin, K. Loken, and M. van de Panne. Simbicon: Simple biped locomotion control. *ACM Transactions on Graphics*, 26(3), 2007.

Glossary

CRBM A Conditional Restricted Boltzmann Machine is an extension of an RBM designed to model time series data. 18, 19

discriminative model Discriminative models typically model a conditional distribution of target outputs given a set of inputs. Discriminative models differ from generative models in that they do not allow one to generate samples from the joint distribution over inputs and outputs (and/or hidden variables). Discriminative models are particularly well suited for input-output tasks such as classification or regression.. 19, 29

filtering distribution The filtering distribution is a distribution of the form $p(X_k|Y_0, Y_0, ..., Y_k)$. 3, 30

generative model Generative models are models capable of generating (synthesizing) observable data. Generative models are able to model joint probability distributions over the input, output and hidden variables in the model. During inference generative models are often used as an intermediate step in forming conditional distribution of interest. Generative models, in contrast to discriminative models, provide a full probabilistic model over all variables, whereas a discriminative model provides a model over the target output variable(s) conditioned on the input variables.. 19, 29

GP A Gaussian Process is a continuous stochastic process defined on a real-valued domain (*e.g.*, time). It defines a Gaussian distribution over functions, and is fully characterized by a mean function and a covariance function. In addition any realization at a finite set of points in the domain (*e.g.*, time instants) form a multivariate Gaussian density. 8–10, 15, 16

GPDM A Gaussian Process Dynamical Model is an extension of the GPLVM to handle high-dimensional time series data. In addition to the probabilistic generative mapping from latent positions to the observation in the GPLVM, it includes a dynamical model that models the temporal evolution of the data in terms of a latent dynamical model. 12–15

GPLVM A Gaussian Process Latent Variable Model is a probabilistic generative model that is learned from high-dimensional data. It can be used as a probabilistic dimensionality reduction, where the latent variables capture the structure (latent causes) of the high-dimensional training data. It is a generalization of probabilistic PCA to nonlinear mappings. 8, 10–15, 19, 29

image edge Image edges are defined as pixels in the image where there exists a discontinuities in the pixel brightness. Image edges are common features used in vision as they are easy to compute and are largely invariant to lighting. 2

Kalman filter Kalman filter is an algorithm for efficiently doing exact inference in a linear dynamical system (LDS), where all latent and observed variables have a Gaussian (or multivariate Gaussian) distribution. 3, 21

LDS A Linear Dynamical System is used to refer to a linear-Gaussian Markov process. In such a process the state evolution is modeled as a linear transformation plus Gaussian process noise. A first-order LDS on state $\mathbf{x}$, for matrix $\mathbf{A}$, is given by $\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \eta$ where η is a Gaussian random variable that is independent of $\mathbf{x}$ and IID through time. 4, 6, 17, 30, 31

MAP Acronym for maximum a posteriori estimate. 2, 12, 19

Markov process A Markov process (or Markov chain) is a time-varying stochastic process that satisfies the Markov property. An n^{th}-order Markov process, $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots)$, satisfies $p(\mathbf{x}_t \,|\, \mathbf{x}_1, \cdots, \mathbf{x}_{t-1}) = p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \cdots, \mathbf{x}_{t-n})$. That is, conditioned on the previous n states, the current state is independent of all other previous states. 3, 4

maximum a posteriori In Bayesian statistics, a maximum a posteriori probability (MAP) estimate is defined as a mode of the posterior distribution.. 30

optical flow Optical flow or optic flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. 2

particle filter The particle filters, also known as sequential Monte Carlo methods (SMC), approximate the posterior filtering distribution with a set of typically weighted samples. 7, 19, 22

PCA Principal Component Analysis is a method for dimensionality reduction, wherein high-dimensional data are projected onto a linear subspace with an orthogonal matrix. It can be formulated as the orthogonal linear mapping that maximizes the variance of projection in the subspace. Probabilistic PCA is a closely related latent variable model that specifies a linear-Gaussian generative process. 5–8, 10, 17, 30

posterior Posterior probability of a random event is the conditional probability once all the relavent evidence is taken into account. According to Bayesian statistical theory posterior can be exprezssed as a product of the the prior and likelihood, *i.e.*, $p(x|I) \propto p(I|x)p(x)$. 2

RBM A Restricted Boltzmann Machine is a bipartite, undirected, probabilistic graphical model. The graph comprises ”visible” (observed) nodes (e.g., image pixels) and ”hidden” (or latent) nodes. The basic RBM has binary random variables, but it has been extended to the real-valued case. The model is restricted in that no edges connect visible or hidden nodes to one another. Rather, all edges connect visible nodes to hidden nodes. Thus, conditioned on the hidden state, the visible variables are independent, and vice versa. This enables efficient learning and inference. 18, 19, 29

SLDS A Switching Linear Dynamical System is a collection of N LDS models along with a discrete switching variable, $s \in \{1, \cdots, N\}$. The switching variable identifies which LDS should be active at each time step. As a probabilistic generative model, each LDS is a linear-Gaussian model, and on maintains a multinomial distribution for s. SLDS models are used to approximate nonlinear dynamical processes in terms of piecewise linear state evolution. 17, 19