Priors for People Tracking from Small Training Sets[*]

Raquel Urtasun
CVLab
EPFL, Lausanne
Switzerland

David J. Fleet
Computer Science Dept.
University of Toronto
Canada

Aaron Hertzmann
Computer Science Dept.
University of Toronto
Canada

Pascal Fua
CVLab
EPFL, Lausanne
Switzerland

Abstract

We advocate the use of Scaled Gaussian Process Latent Variable Models (SGPLVM) to learn prior models of 3D human pose for 3D people tracking. The SGPLVM simultaneously optimizes a low-dimensional embedding of the high-dimensional pose data and a density function that both gives higher probability to points close to training data and provides a nonlinear probabilistic mapping from the low-dimensional latent space to the full-dimensional pose space. The SGPLVM is a natural choice when only small amounts of training data are available. We demonstrate our approach with two distinct motions, golfing and walking. We show that the SGPLVM sufficiently constrains the problem such that tracking can be accomplished with straighforward deterministic optimization.

1. Introduction

The 3D estimation of human pose from monocular video is often poorly constrained, owing to reflection ambiguities, self-occlusion, cluttered backgrounds, non-rigidity of tissue and clothing, and poor image resolution. As a consequence, prior information is essential to resolve ambiguities, minimize estimator variance, and to cope with partial occlusions. Unfortunately, because of the high-dimensional parameterization of human models, learning prior models is difficult with small or modest amounts of training data. Manual design of suitable models is also very difficult.

This paper describes an effective method for learning prior models from training data comprising typical body configurations, and then using them for 3D people tracking. We exploit the recently developed Scaled Gaussian Process Latent Variable Model (SGPLVM [6, 9]) to learn a low-dimensional embedding of high-dimensional human pose data. The model can be learned from much smaller amounts of training data than competing techniques (e.g., [5, 18]), and it involves very few manual tuning parameters.

SGPLVM provides a continuous, kernel-based density function $p(\mathbf{x}, \mathbf{y})$ over positions $\mathbf{x}$ in a low-dimensional latent space and positions $\mathbf{y}$ in the full pose space. The density function is generally non-Gaussian and multimodal. Importantly, it provides a natural preference for poses close to the training data, smoothly falling off with distance. The model also provides a simple, nonlinear, probabilistic mapping from the latent space to the full pose space. As explained below, $\mathbf{y}$ conditioned on $\mathbf{x}$ is a Gaussian random variable. Its variance reflects the uncertainty of the mapping, and therefore increases with the dissimilarity between $\mathbf{x}$ and the training data. This explicit representation of the variance is extremely useful.

This paper explores the use of SGPLVM priors for monocular 3D people tracking. To this end we consider two distinct domains, golfing and walking. Priors are learned from a single exemplar of each motion class. Both examples last just a second but are, nevertheless, shown to be sufficient for tracking. The generative model for the tracker comprises the SGPLVM, a simple likelihood function, and a second-order dynamical model. Online tracking is accomplished with straightforward deterministic optimization.

2 Previous Work

A barrier to learning useful prior models for 3D human pose and motion stem from the high dimensional representations commonly used to describe human pose and motion. A second barrier is the relative sparsity of natural human poses within such a pose space.

Some of the earliest attempts to learn more effective, low-dimensional models for visual tracking used principal component analysis (e.g., [3, 23, 16]). Linear subspace models, while useful for some tracking problems, are inappropriate for the nonlinear, multimodal space of typical human poses. Gaussian mixture models and mixtures of factor analyzers are better able to handle manifolds and multimodal densities (e.g., [7]), but they suffer in high dimensional pose spaces where they tend to overfit the data unless a very large set of training data is used. In particular, the number of parameters quickly becomes untenable, and it

[*] This work was supported in part by the Swiss National Science Foundation, by NSERC Canada and by the Canadian Institute for Advanced Research.

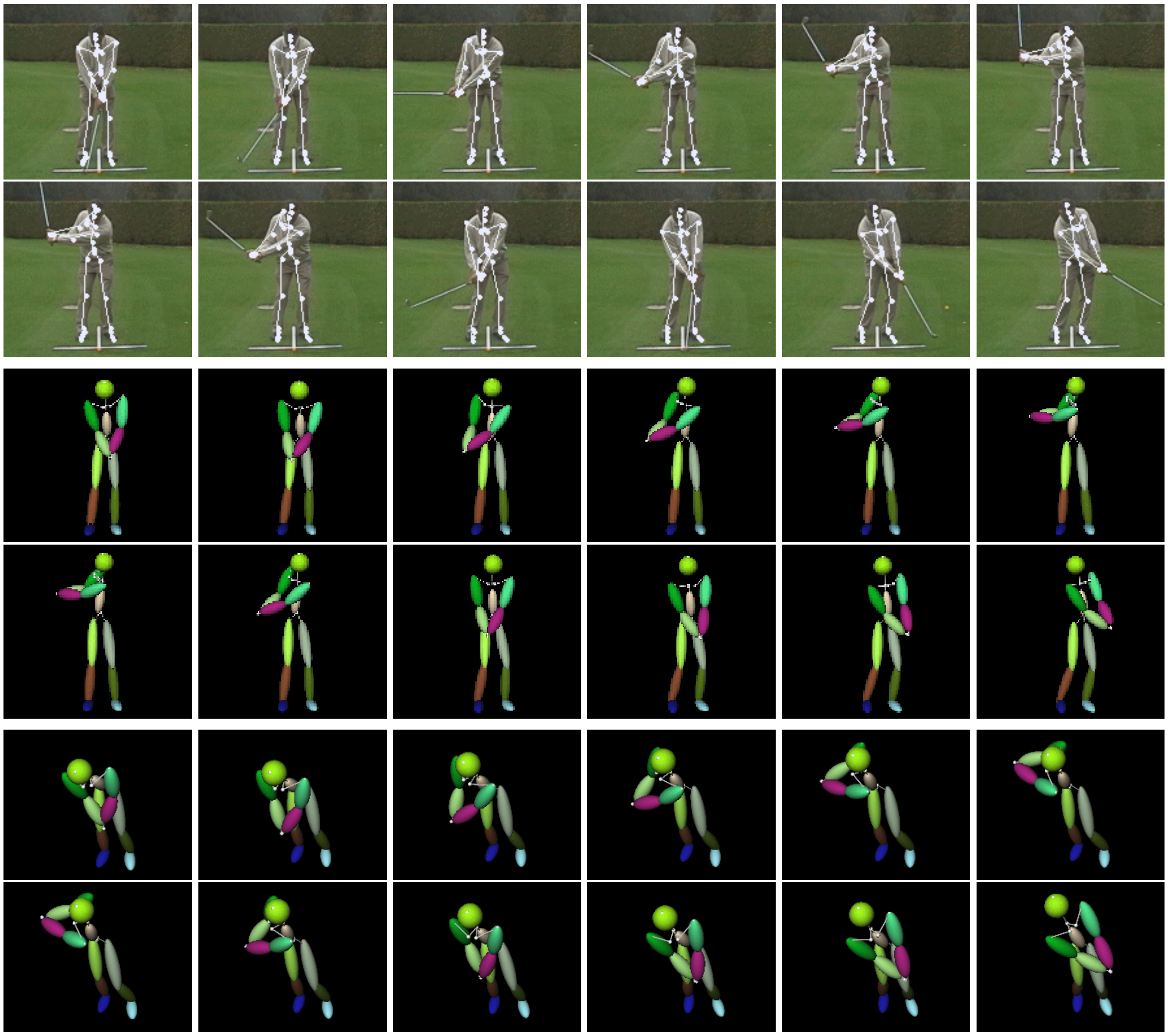

Figure 1. Tracking of a 62-frame *short* golf swing. **Top two rows**: The skeleton of the recovered 3D model is projected into a representative subset of images. **Middle two rows**: Volumetric primitives of the recovered 3D model projected into the same views. **Bottom two rows**: Volumetric primitives of the 3D model as seen from above.

can be extremely difficult to choose a reasonable number of (Gaussian) component densities.

Non-parametric models can be effective, but they often require large amounts of training data, especially in a high-dimensional pose space [11, 17]. Even with large amounts of training data, they can be problematic because they do not produce a smooth density function. As a consequence, one can only infer poses in the training set.

One alternative is to expoit nonlinear dimensionality reduction methods to construct a low-dimensional embedding of the training data [5, 14, 18, 24]. However, while LLE (local linear embedding) [15], Isomap [20], and spectral embedding techniques [2, 18] provide a low-dimensional representation of the data, they do not produce a density model in the embedding space, nor do they provide straightforward mappings between the embedding space and the full pose space. Of course, one can first learn the embedding, and then learn a density model and the inverse mapping. For example Wang et al. [24] use Isomap to learn the embedding. Then they assume a mixture of factor analyzers and an approximate linear model based on K-nearest neighbors to learn a latent density and a mapping to the full state space. This mapping will generally be discontinuous and is therefore inappropriate for continuous optimization. Sminchisescu and Jepson [18] use spectral embedding for 3D pose data. Given the embedding, they learn a Gaussian mixture model as a density model for the training data in the embedding space, and then a mapping from the embedding to the pose space using RBF regression. To learn a prior model for walking they used several thousand training poses.

Here, we use a Scaled Gaussian Process Latent Variable Model (SGPLVM) that learns a generative model with a continuous mapping between the latent space and the full pose space, even for very small training sets [9]. Grochow et al. [6] introduced the use of a SGPLVM of human pose for interactive computer animation. However, this required significant human interaction for pose inference and interpolation. More recently, Tian et al. [21] used a GPLVM to constrain the estimation of 2D upperbody pose from 2D silhouettes.

3. Gaussian Process Models

Gaussian processes are often introduced in the context of regression, to learn a mapping $\mathbf{y} = f(\mathbf{x})$ from training pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$ [13]. In least-squares regression, the quality of the result often depends greatly on the specific form of f that one fits. Gaussian processes arise from a Bayesian formulation in which one marginalizes over a family of functions for f. In this way, one mitigates common problems due to overfitting and underfitting. One can additionally learn the smoothness and noise parameters. Remarkably, for a wide class of functions, this marginalization produces a Gaussian process model [13].

3.1. SGPLVM

In contrast to regression problems, the GPLVM [9] and SGPLVM [6] likelihoods of the training data points $\{\mathbf{y}_i\}_{i=1}^N$, $y_i \in \mathcal{R}^D$, are modeled as Gaussian processes for which the corresponding values $\{\mathbf{x}_i\}$ are initially unknown. As a consequence, one must now learn the unknown latent positions $\{\mathbf{x}_i\}$ along with the mapping from $\mathbf{x}$ to $\mathbf{y}$. This formulation can be viewed as a generalization of probabilistic PCA [22] where, instead of marginalizing over latent variables $\mathbf{x}$ to find the linear mapping from $\mathbf{x}$ to $\mathbf{y}$, we marginalize over mapping functions and optimize the latent positions $\{\mathbf{x}_i\}$. A kernel function is introduced to allow for nonlinear mappings [9]. Scaling of individual data dimensions was introduced by Grochow et al. [6] to account for different variances of different dimensions of the data.

More formally[1], let $\mathbf{Y} \equiv [\mathbf{y}_1, \cdots, \mathbf{y}_N]^T$ be a matrix, each row of which is one of the training data points. We assume that the mean $\mu \in \mathcal{R}^D$ has been subtracted from the data, so that the $\mathbf{y}_i$ are mean zero. Under the Gaussian process model, the conditional density for the data is multivariate Gaussian

$$p(\mathbf{Y} \mid M) = \frac{|\mathbf{W}|^N}{\sqrt{(2\pi)^{ND}|\mathbf{K}|^D}} \exp(-\frac{1}{2}\,\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{W}^2\mathbf{Y}^T)) , \tag{1}$$

where $M \equiv \{\{\mathbf{x}_i\}, \alpha, \beta, \gamma, \{w_j\}_{j=1}^D\}$ are the unknown model parameters, $\mathbf{W} \equiv \mathrm{diag}(w_1, ..., w_D)$ is a diagonal matrix containing a scaling factor for each data dimension, and $\mathbf{K}$ is a kernel matrix. The elements of the kernel matrix are given by a kernel function, $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Following [6, 9], we use a RBF kernel function, with parameters α, β and γ, given by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha \exp(-\frac{\gamma}{2}||\mathbf{x}_i - \mathbf{x}_j||^2) + \beta^{-1}\delta_{\mathbf{x}_i,\mathbf{x}_j} \ , \tag{2}$$

where $\delta_{\mathbf{x}_i,\mathbf{x}_j}$ is the Kronecker delta function.

During training, we learn the model parameters M by minimizing the negative log posterior $-\ln p(M \mid \mathbf{Y})$. Following [9] this posterior comprises the likelihood in (1), a simple prior on the kernel hyperparameters, and an isotropic IID Gaussian prior on the latent positions. In minimizing the log posterior we simultaneously learn the latent positions corresponding to the training data points, along with a continuous mapping from the latent space to the full pose space. By contrast, with other techniques (e.g., [5, 18, 24]) the embedding is specified first and a mapping is then learned separately.

Once the model parameters M are learned, the joint density over a new latent position $\mathbf{x}$ and an associated pose $\mathbf{y}$ is given by [13]

$$\begin{aligned} p(\mathbf{x}, \mathbf{y} \mid M, \mathbf{Y}) \quad &\propto \quad \exp(-\frac{\mathbf{x}^T\mathbf{x}}{2}) \frac{|\mathbf{W}|^{N+1}}{\sqrt{(2\pi)^{(N+1)D}|\hat{\mathbf{K}}|^D}} \\ &\times \ \exp(-\frac{1}{2}\,\mathrm{tr}(\hat{\mathbf{K}}^{-1}\hat{\mathbf{Y}}\mathbf{W}^2\hat{\mathbf{Y}}^T)) \end{aligned} \tag{3}$$

where $\hat{\mathbf{Y}} \equiv [\mathbf{y}_1, \cdots, \mathbf{y}_N, \mathbf{y}]^T$ comprises the training data points and the new pose $\mathbf{y}$, and $\hat{\mathbf{K}}$ is the corresponding new kernel matrix:

$$\hat{\mathbf{K}} = \begin{pmatrix} \mathbf{K} & \mathbf{k}(\mathbf{x}) \\ \mathbf{k}(\mathbf{x})^T & k(\mathbf{x}, \mathbf{x}) \end{pmatrix} , \tag{4}$$

where $\mathbf{k}(\mathbf{x}) \equiv [k(\mathbf{x}_1, \mathbf{x}), \cdots, k(\mathbf{x}_N, \mathbf{x})]^T$.

Following [6, 13], one can derive a more useful expression for the likelihood of a new pair $(\mathbf{x}, \mathbf{y})$. That is, up to an additive constant, the negative log probability, $-\ln p(\mathbf{x}, \mathbf{y} \mid M, \mathbf{Y})$, is equal to

$$L(\mathbf{x}, \mathbf{y}) = \frac{||\mathbf{W}(\mathbf{y} - \mathbf{f}(\mathbf{x}))||^2}{2\sigma^2(\mathbf{x})} + \frac{D}{2}\ln\sigma^2(\mathbf{x}) + \frac{1}{2}||\mathbf{x}||^2 , \tag{5}$$

with

$$\mathbf{f}(\mathbf{x}) \quad = \quad \mu + \mathbf{Y}^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}) , \tag{6}$$

$$\sigma^2(\mathbf{x}) \quad = \quad k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}) . \tag{7}$$

Here, $\mathbf{f}(\mathbf{x})$ is the mean pose reconstructed from the latent position $\mathbf{x}$, i.e., the mean of $p(\mathbf{y} \mid \mathbf{x}, M, \mathbf{Y})$. Using (6), the mapping from the latent space to the pose space is continuous and relatively simple to compute. The variance, $\sigma^2(\mathbf{x})$, gives the uncertainty of the reconstruction; it is expected to be small in the vicinity of the training data, and large far from them. Therefore, minimizing $L(\mathbf{x}, \mathbf{y})$ aims to minimize reconstruction errors (i.e., to keep $\mathbf{y}$ close to $\mathbf{f}(\mathbf{x})$), while keeping latent positions close to the training data (i.e., to keep $\sigma^2(\mathbf{x})$ small). The third term in (5) is the result of a broad prior over latent positions that usually has relatively little influence on the optimized latent positions.

[1] We refer the interested reader to [6, 9, 13] for more details.

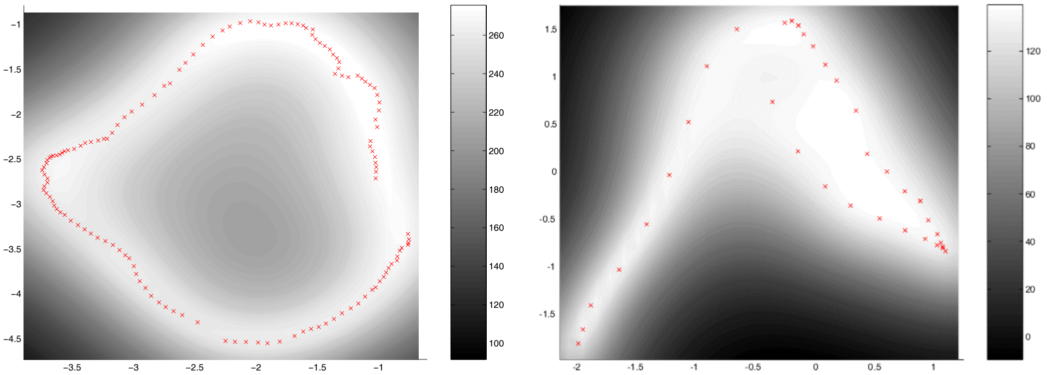

Figure 2. 2D SGPLVM latent space $\mathbf{x}$ learned from motion capture data. The grayscale plot represents $-\frac{D}{2}\ln\sigma^2(\mathbf{x})-\frac{1}{2}||\mathbf{x}||^2$. The red crosses are the optimized $\mathbf{x}_i$ positions associated with the training poses. **Left**: Using 139 frames of a walking cycle performed on a treadmill and retaining 24 active set points. **Right**: Using 35 frames of a golf swing and retaining 19 active set points.

3.2. Active Set

The main computational burden with a Gaussian process is the inversion of the $N \times N$ kernel matrix, where N is the number of data samples. We also wish to limit the size of $\mathbf{K}$ in order to obtain a sufficiently smooth prior (avoiding overfitting). Following [9, 10], while learning is based on the entire training set, the SGPLVM is constructed from a subset of the data referred to as the *active set*. In a greedy fashion, data points are added to the model one point at a time; at each step one chooses the point with the highest reconstruction variance (7). In this way, the active set tends to include training data points that are reasonably well spaced throughout the latent space.[2]

3.3. Learning Specific Motions

In our specific application, each training point $\mathbf{y}_i$ is a vector of joint angles that describes a body pose. We do not include global position and orientation of the torso in this pose vector. We represent the human body as an articulated structure with 84 degrees of freedom for walking and 72 for golfing. These numbers are those in the training databases over which we had no control. While careful choice of joint representations can have a large influence on the success of parameter estimation and tracking, here we simply used the data as provided in the motion capture databases.

Figure 2 shows the 2D latent space learned for the walking and golfing motions. The greyscale plot represents $-\frac{D}{2}\ln\sigma^2(\mathbf{x})-\frac{1}{2}||\mathbf{x}||^2$, thus depicting the regions of latent space that produce more likely poses. The walking model was learned from a single walk cycle performed on a treadmill, comprising 139 poses obtained with an optical motion capture system. 24 poses were chosen automatically for the active set. Even though the walking cycle was not symmetrized, as indicated by the gap between the beginning and end of the gait cycle on the right side of Fig. 2 (left), the SGPLVM effectively completes the curve with a low variance region that fills the gap. The golfing model was learned from a single swing composed of 35 poses taken from the CMU database [4]. The active set contains 19 points and produces the smooth model shown in Fig. 2 (right). For both walking and golfing, we used a 2D latent space, in part for simplicity and to allow for periodic motions.

4. Monocular Tracking

The complete body pose is controlled by a state vector $\phi_t = [\mathbf{G}_t, \mathbf{y}_t, \mathbf{x}_t]$, where $\mathbf{G}_t$ represents the global position and orientation of the body, $\mathbf{y}_t$ the body pose, and $\mathbf{x}_t$ the coordinates in the latent space. Given an image sequence $\mathbf{I}_{1:t} \equiv (\mathbf{I}_1, ..., \mathbf{I}_t)$, and the learned model M, we formulate the tracking problem as one of maximizing $p(\phi_t \,|\, \mathbf{I}_{1:t}, M, \mathbf{Y})$. That is, we wish to find the MAP pose estimate at each time, denoted ϕ_t^{MAP}. Assuming Markov dynamics and conditional independence of the observations, we write the well-known filtering distribution as

$$p(\phi_t \,|\, \mathbf{I}_{1:t}, M, \mathbf{Y}) \;\propto\; p(\mathbf{I}_t \,|\, \phi_t)\, p(\phi_t \,|\, \mathbf{I}_{1:t-1}, M, \mathbf{Y}) \;. \quad (8)$$

Here, $p(\mathbf{I}_t \,|\, \phi_t)$ is the likelihood of the current measurements conditioned on the pose, and $p(\phi_t \,|\, \mathbf{I}_{1:t-1}, M, \mathbf{Y})$ is the density over poses predicted by previous measurements and the learned SGPLVM.

We further assume that the prediction distribution $p(\phi_t \,|\, \mathbf{I}_{1:t-1}, M, \mathbf{Y})$ can be factored into two components, one that prefers poses close to the training data, and one that prefers smooth motions; i.e.,

$$p(\phi_t \,|\mathbf{I}_{1:t-1}, M, \mathbf{Y}) \approx p(\phi_t \,|M, \mathbf{Y})\, p(\phi_t \,|\phi_{t-1}^{MAP}, \phi_{t-2}^{MAP}) \quad (9)$$

To encourage poses to be close to the training data, we use the SGPLVM. The corresponding log prior over poses is simply given by the log likelihood in (5), i.e.,

$$-\ln p(\phi_t \,|\, M,\, \mathbf{Y}) = L(\mathbf{x}, \mathbf{y}) \;. \quad (10)$$

[2] See [9] for details concerning the active set and heuristics used during learning and http://www.dcs.shef.ac.uk/ neil/gplvm/ for the GPLVM code.

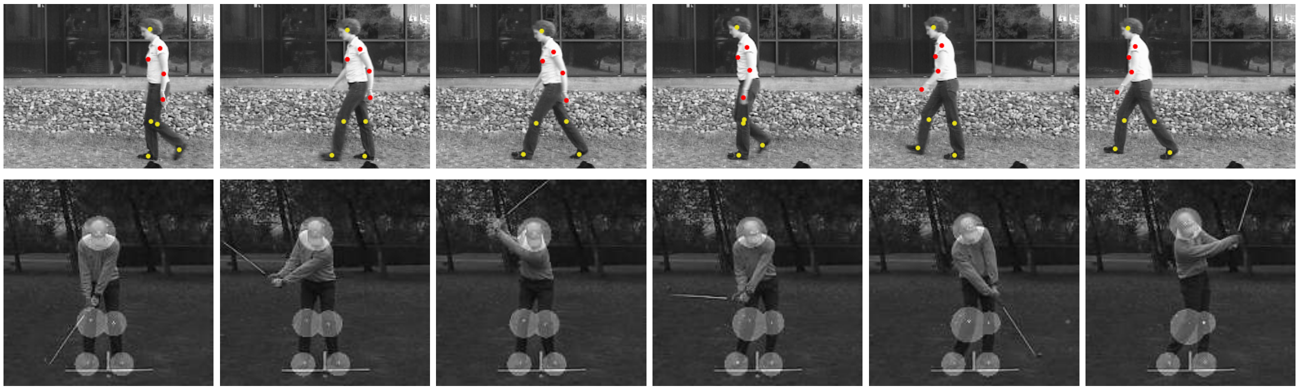

Figure 3. 2D Tracking using the WSL tracker. **Top row:** Tracking the chest, knees, head, ankles and visible arm. The tracked upper body joints are shown in red, with the head and tracked lower joints points shown in yellow. **Bottom row**: Regions used for tracking the ankles, knees, and head are shown.

To encourage smoothness, we assume a second-order Gauss-Markov model over the global orientation and position, $\mathbf{G}_t$, and the joint angles $\mathbf{y}_t$. Given MAP estimates from the previous two time instants, the negative log transition density is, up to an additive constant,

$$-\ln p(\phi_t \,|\, \phi_{t-1}^{MAP}, \phi_{t-2}^{MAP}) \;=\; \frac{||\mathbf{y}_t - \hat{\mathbf{y}}_t||^2}{2\sigma_y^2} + \frac{||\mathbf{G}_t - \hat{\mathbf{G}}_t||^2}{2\sigma_G^2} \quad (11)$$

where the mean predictions $\hat{\mathbf{G}}_t$ and $\hat{\mathbf{y}}_t$ are

$$\hat{\mathbf{y}}_t \;=\; 2\mathbf{y}_{t-1}^{MAP} + \mathbf{y}_{t-2}^{MAP} \quad , \quad \hat{\mathbf{G}}_t \;=\; 2\mathbf{G}_{t-1}^{MAP} + \mathbf{G}_{t-2}^{MAP} \; .$$

The standard deviations were deliberately set large as it was often the case that the dynamical model did not play a critical role in the optimization.

The image observations used for the 3D tracker were the approximate 2D image locations of a small number (J) of joints (see Fig. 3). They were obtained with a 2D image-based tracker. The likelihood function is derived by assuming mean-zero Gaussian noise in the 2D measurements provided by the 2D tracker. Let the perspective projection of the j^{th} joint position, $\mathbf{p}^j$, in pose ϕ_t, be denoted $P(\mathbf{p}^j(\phi_t))$, and let the associated 2D image measurement from the tracker be $\hat{\mathbf{m}}_t^j$. Then, the negative log likelihood of the observations given the state is

$$-\ln p(\mathbf{I}_t \,|\, \phi_t) \;=\; \frac{1}{2\sigma_e^2} \sum_{j=1}^{J} \left\| \hat{\mathbf{m}}_t^j - P(\mathbf{p}^j(\phi_t)) \right\|^2 \; , \quad (12)$$

The standard deviation was set to $\sigma_e = 3$ based on empirical results with the 2D trackers used.

For example, Fig. 3 shows the 2D tracking locations for two test sequences. With the walking sequence we tracked 9 points on the body. For the golfing sequences we used 6 points. The fact that we use such a small number of tracked points is notable. By comparison, most successful 3D people trackers exploit several sources of image information, including edges, flow, silhouettes, skin detection, etc. The small number of constraints is also remarkable when compared to the dimension of the training poses. While it is known that 2D joint locations are useful for 3D pose estimation (e.g., [19]), it would not be possible for the optimization to find the pose parameters without a suitable prior.

Finally, MAP estimates were obtained using straightforward deterministic optimization. In particular, we minimize the negative log posterior obtained by substituting (9) into (8). After initializing the 2D tracker in the first frame, the optimization is performed online, one frame at a time. The initial state for the optimization at each frame is given by the mean of the transition density, $\hat{\mathbf{y}}_t$ and $\hat{\mathbf{G}}_t$. Given that initial pose $\hat{\mathbf{y}}_t$, we first obtain an initial latent position $\hat{\mathbf{x}}_t = \arg\min_{\mathbf{x}} L(\mathbf{x}, \hat{\mathbf{y}}_t)$. Using this initial guess, we then use standard optimization techniques to minimize the log posterior in (8), thereby finding the desired MAP estimate.

5. Results

The results shown in this paper were obtained from uncalibrated images. The motions were performed by subjects of unknown sizes wearing ordinary clothes that are not particularly textured. To perform our computation, we used rough guesses for the subject sizes and for the intrinsic and extrinsic camera parameters to match the 2D projections.

5.1. Walking Motion

Figure 6 shows a well-known walking sequence. For 2D tracking we used the WSL tracker [8]. WSL is a robust, motion-based 2D tracker that maintains an online adaptive appearance model. The model adapts to slowly changing image appearance with a natural measure of the temporal stability of the underlying image appearance. It also permits the tracker to handle partial occlusions.

For each test sequence we manually initialized the 3D position and orientation of the root node of the body, $\mathbf{G}$, in the first frame so that it projects approximately to the

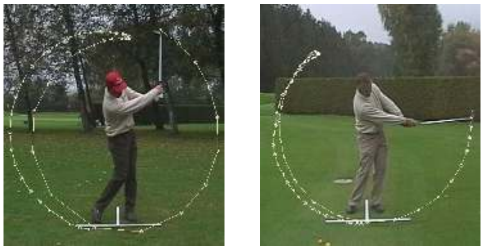

Figure 4. Detected club trajectories for the *full* swing of Fig. 7 and the *short* swing of Fig. 1. Note that the full swing has a much longer trajectory than the other.

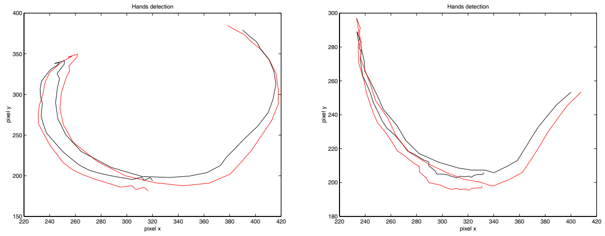

Figure 5. Detected hand trajectories for *full* swing in Fig. 7 and the *short* swing in Fig. 1. Left and right hand positions (pixel units) are represented in black and red respectively.

right place. Similarly we manually gave the 2D locations of a few joints to be tracked by WSL. As depicted in Fig. 3 (top row), 9 joints were tracked, namely, the ankles, knees, chest, head, left shoulder, elbow and hand. This entire process only required a few mouse clicks and could easily be automated using posture detection techniques [1, 5]. The initial states for the dynamical model, ϕ_0 and ϕ_1, were chosen to be those in the training database that best projected onto the first two frames.

Figure 6 shows the estimated 3D model projected onto several frames of the sequence, as well as some rendered 3D volumetric models. Note how well the skeleton reprojects onto the limbs even though the motion was learned from a single cycle of a different person on a treadmill. It is also interesting to see how well the arm is tracked (cf. [16]).

5.2. Golf Swing

As discussed in Section 3.3, the golf swing used to train the SGPLVM was a *full swing* from the CMU motion database [4]. It was performed by neither of the two golfers used for tracking (see Figs. 1 and 7). Here, we tracked five points using the WSL tracker, namely the knees, ankles and head. The initialization of these points could be also automated using posture detection techniques since the pose at the beginning of the swing is quite stereotyped.

Because the hands tend to rotate during the motion, to track the wrists we have found it effective to use a club tracking algorithm [12] that takes advantage of the information provided by the whole shaft. Its output is depicted in Fig. 4. This tracker does not require any manual initialization. It is also robust to mis-detections and false alarms and has been validated on many sequences. From the recovered club motion, we can infer the 2D hand trajectories as shown in Fig. 5.

The first two rows of Fig. 7 depict the projections of the recovered 3D skeleton in a representative subset of images of a *full* swing. The bottom two rows show projections of the 3D model using a viewpoint similar to the one of the original camera.

Fig. 1 depicts a *short* swing that is performed by a different person. Note that this motion is quite different both from the full swing motion of Fig. 7 and from the swing used to train the SGPLVM. The club does not go as high and, as shown in Fig. 5, the hands travel a much shorter distance. The tracking nevertheless remains very accurate. This helps illustrate the usefulness of the SGPLVM generalization.

In Fig. 8, we compare our current results with those obtained with a prior motion model learned with PCA from all ten swings in the CMU motion database. In that tracker [23] we used the same 2D tracked points, along with a brightness constancy constraint and 2D silhouette information. It also required pose detection of keyframes for initialization. By comparison, the method here is entirely online. Notice that with the SGPLVM the skeleton's projection matches the limbs better than with the linear PCA-based model.

6. Summary and Conclusions

We presented a SGPLVM-based method to learn prior models of 3D human pose and showed that it can be used effectively for monocular 3D tracking. In the case of both walking and golfing, we have been able to recover the motion from video sequences given a single examplar of each motion to train the model. Furthermore, these priors sufficiently constrain the problem so that this could be accomplished with straightforward deterministic optimization. This is in sharp contrast to competing techniques that either involve large amounts of training data or computationally expensive multi-hypothesis tracking algorithms.

In this paper, tracking was accomplished with particularly simple second-order dynamics and an observation model based on a very small number of tracked features. The quality of our results with such simple dynamics and appearance models clearly demonstrates the power of the SGPLVM prior models. More sophisticated appearance and dynamics models should produce even better results. Further work will focus on incorporating dynamics into the priors to increase robustness, and on incorporating multiple motion classes and transitions between them.

References

[1] A. Agarwal and B. Triggs. 3D human pose from silhouettes by relevance vector regression *Proc. CVPR*, Vol.2 pp. 882-888, Washington, 2004

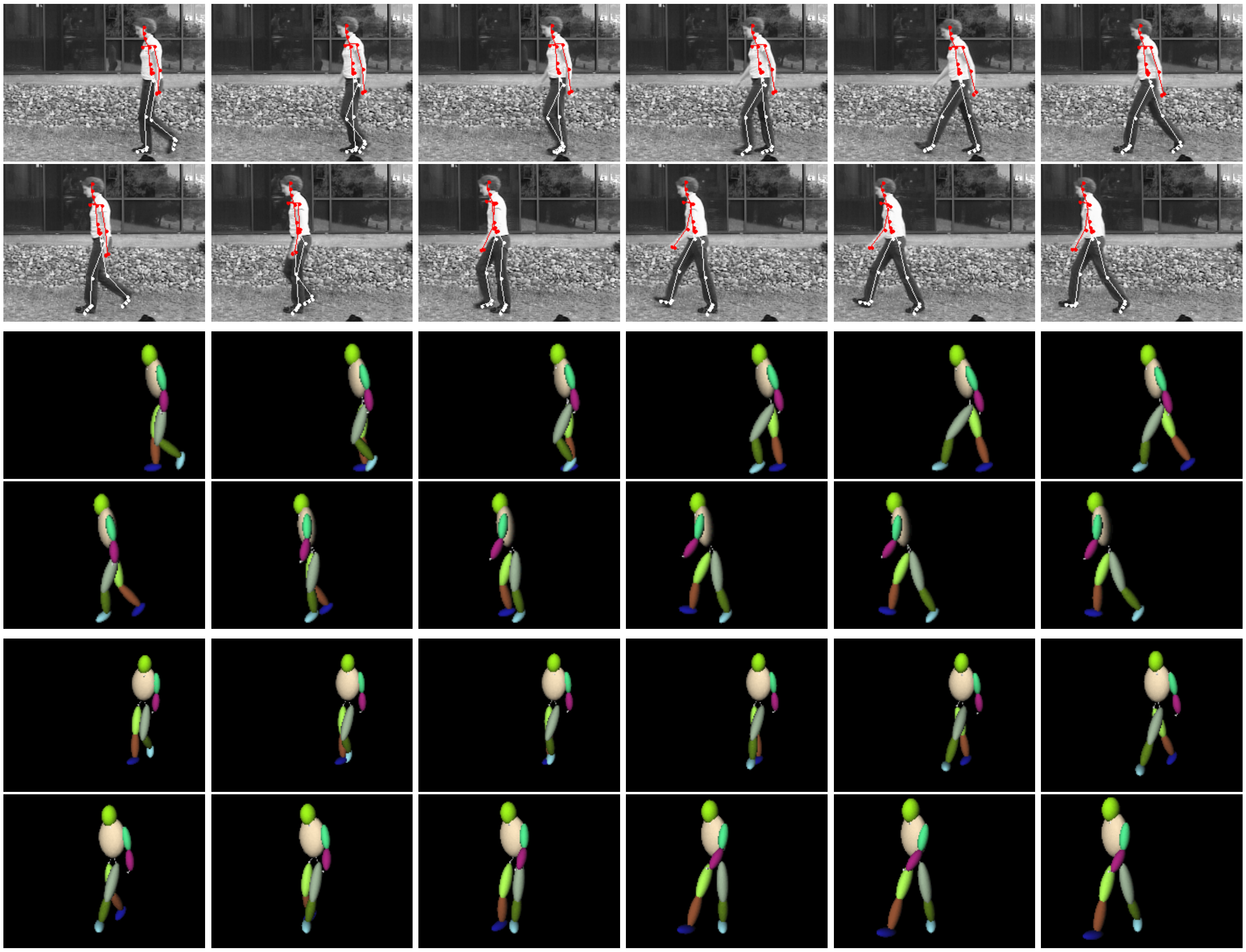

Figure 6. Tracking 32 frames of a walking motion. **First two rows**: The skeleton of the recovered 3D model is projected onto the images. **Middle two rows**: Volumetric primitives of the recovered 3D model projected into a similar view. **Bottom two rows**: Volumetric primitives of the 3D model as seen from the front view.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *NIPS 14*, pp. 585-591.

[3] M. J. Black and A. D. Jepson. EigenTracking: Robust matching and tracking of articulated objects using a view-based representation. *IJCV*, 26(1):63-84, 1998.

[4] CMU database. http://mocap.cs.cmu.edu/.

[5] A. Elgammal and C.S. Lee. Inferring 3D body pose from silhouettes using activity manifold learning. *Proc. CVPR*, Vol. 2 pp. 681-688, Washington, 2004.

[6] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. *Proc. SIGGRAPH*, pp. 522-531, 2004.

[7] N. R. Howe, M. E. Leventon, and W. T. Freeman. Bayesian reconstructions of 3D human motion from single-camera video. *NIPS 12*, pp. 281-288, MIT Press, 1999.

[8] A.D. Jepson, D. J. Fleet, and T. El-Maraghi. Robust on-line appearance models for vision tracking. *IEEE Trans. PAMI*, 25(10):1296-1311, 2003.

[9] N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *NIPS 16*, pp. 329-336 MIT Press, 2004.

[10] N. D. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. *NIPS 15*, pp. 609–616, MIT Press, 2003.

[11] J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard. Interactive control of avatars animated with human motion data. *Proc. SIGGRAPH*, pp. 491-500, 2002.

[12] V. Lepetit and A. Shahrokni and P. Fua. Robust data association for online applications. *Proc. CVPR*, Vol. 1 pp. 281-288, Madison, 2003.

[13] D. J. C. MacKay. Introduction to Gaussian processes. In *Neural Networks and Machine Learning*, C. Bishop, ed., NATO ASI Series, pp. 133-166. Kluwer, 1998.

[14] A. Rahimi, B. Recht, T. Darrell. Learning appearance manifolds from video. *CVPR*, v1, pp. 868-875, San Diego, 2005.

[15] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323-2326, 2000.

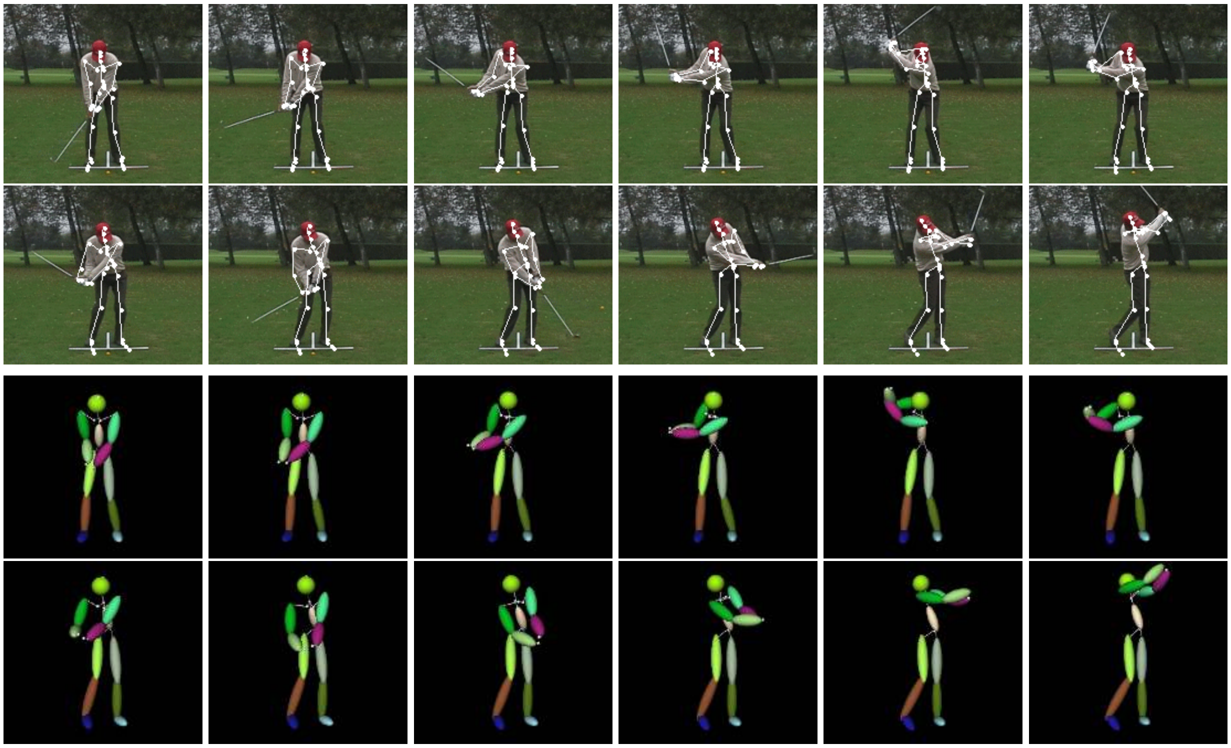

Figure 7. Tracking of a *full* golf swing in a 50 frames sequence. **First two rows**: The skeleton of the recovered 3D model is projected into a representative subset of images. **Last two rows**: Volumetric primitives of the 3D model as seen from viewpoint similar to the camera used.

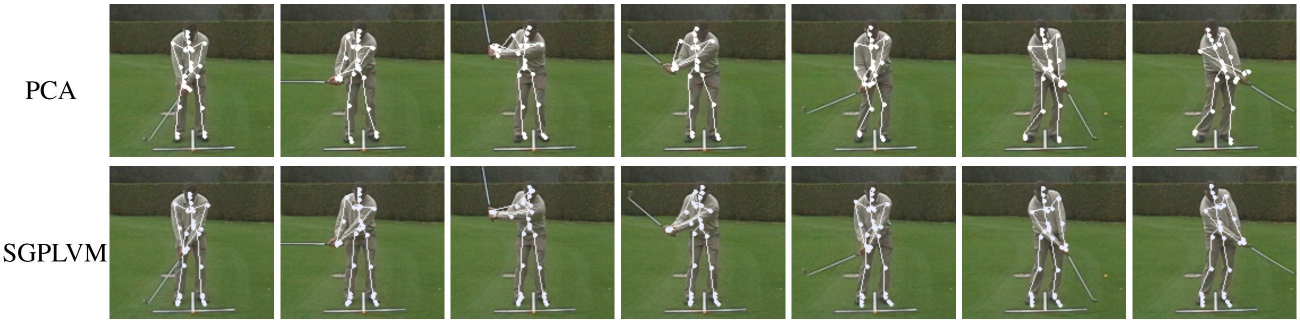

Figure 8. Comparison between the PCA-based tracker [23] in the **first row** and the SGPLVM-based tracker in the **second row**. Note the gain in accuracy due to the generalization ability of the SGPLVMs.

[16] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3D human figures using 2D image motion. *Proc. ECCV*, pp. 702-718, Dublin, 2000.

[17] H. Sidenbladh, M. J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. *Proc. ECCV*, pp. 784-800, Copenhagen, 2002.

[18] C. Sminchisescu and A. Jepson. Generative modeling for continuous non-linearly embedded visual inference. *Proc. ICML*, Banff, July 2004.

[19] C.J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *CVIU*, 80:349-363, 2000.

[20] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319-2323, 2000.

[21] T. Tian, R. Li and S. Sclaroff. Articulated pose estimation in a learned smooth space of feasible solutions. Worshop on Learning in Comp. Vis. & Pattern Recog., San Diego, 2005.

[22] M.E. Tipping and C.M. Bishop. Probabilistic principal component anlaysis. *J. Royal Stat. Soc., B*, 6(3):611-622, 1999.

[23] R. Urtasun, D. Fleet and P. Fua. Monocular 3D tracking of the golf swing. *CVPR*,Vol. 2 pp. 932-938, San Diego, 2005.

[24] Q. Wang, G. Xu, and H. Ai. Learning object intrinsic structure for robust visual tracking. *Proc. CVPR*, Vol. 2 pp. 227-233, Madison, 2003.