Functional Analysis of Human Motion Data*

Dirk Ormoneit[†], **Trevor Hastie**[†], and **Michael Black**[‡]

[†] Dept. of Statistics, Stanford University, Stanford, CA 94305

{ormoneit,hastie}@stat.stanford.edu, http://www-stat.stanford.edu/

[‡] Xerox Palo Alto Research Center, Palo Alto, CA 94304

black@parc.xerox.com, http://www.parc.xerox.com/black/

Abstract

We present a method for the modeling of 3D human motion data using functional analysis. First, we estimate a statistical model of typical activities from a large set of 3D human motion data. For this purpose, the human body is represented as a set of articulated cylinders and the evolution of a particular joint angle is described by a time-series. Specifically, we consider periodic motion such as "walking" in this work, and we develop a new set of tools that allows for the automatic segmentation of the training data into a sequence of identical "motion cycles". Then we compute the mean and the principal components of these cycles using a new algorithm that accounts for missing information and that enforces smooth transitions between cycles. As an application of this methodology we consider the visual tracking of human motion in a 2D video sequence. Here the principal components serve to define a low-dimensional representation of the human 3D poses

*Parts of the results described in this work have been presented at the the joint IMS/Beroulli Conference in Guanajuato, Mexico, May 2000 as well as at the IEEE Workshop on Human Modeling, Analysis and Synthesis. Hilton Head Island, South Carolina, June 2000.

in a state-space model that treats the 2D video images as observations. We apply (approximate) Bayesian inference using a particle filter to the state-space model to infer the body poses at each time-step. The resulting algorithm is able to track human subjects in monocular video sequences and to recover their 3D motion in complex unknown environments.

Keywords

Human Motion Data, Functional Data Analysis, Missing Data, Singular Value Decomposition

I. Introduction

We present a new statistical approach to the modeling of 3-dimensional human motion data. Increasing amounts of motion data are available from commercial motion capture systems and are used for a variety of applications. For example, they provide an input for the animation of virtual characters in computer graphics or they can be useful for the analysis of motion abnormalities in medicine. In this work, we focus on an application for the tracking of human motion in 2D video images which is by itself a key issue in many practical problems including human-computer interaction and automatic surveillance. Technically, the human body can be approximated by a collection of articulated limbs (Figure 1) and its motion can be thought of as a collection of time-series describing the joint angles as they evolve over time. A key difficulty for the modeling of these body angles is that each time-series has to be decomposed into suitable temporal primitives prior to statistical

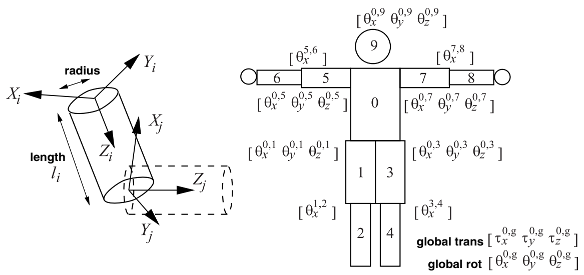

Fig. 1. Human body model, consisting of a set of articulated cylinders with 25 degrees of freedom (DOF). Each limb, i, has a local coordinate system with the Z_i axis directed along the limb. Joints have up to 3 angular DOF, expressed as relative rotations $(\theta_x^{j,i}, \theta_y^{j,i}, \theta_z^{j,i})$ between body parts i and j.

analysis. For example, in the case of repetitive human motion such as walking, motion sequences decompose naturally into a sequence of identical "motion cycles". Of course, the exact nature of this decomposition is unknown to the modeler and needs to be estimated from the motion data. In this work, we present a new set of tools that carry out this identification automatically. In detail, we suggest an iterative procedure that generates the best segmentation with respect to the signal-to-noise ratio of the data in an aligned reference domain. This procedure allows us to use the mean and the principal components of the individual cycles in the reference domain as a statistical model. Technical difficulties include missing information in the motion time-series and the necessity of enforcing smooth transitions between different cycles. To deal with these problems, we develop a new iterative method for functional Principal Component Analysis (PCA) that is based on a Singular Value Decomposition (SVD) in the Fourier domain.

In the tracking application, the principal components serve to define a low-dimensional representation of the human 3D poses in a state-space model that treats the 2D video images as observations. In detail, we specify the transition probabilities of this state-space model based on the singular values of the SVD and we also specify a likelihood model based on a generative model of image appearance. This generative model is highly non-linear so that we have to resort to approximate inference techniques for filtering. Specifically, we apply a particle filter where the target distribution is represented using a discrete set of samples and is propagated over time. The resulting algorithm is able to track human subjects in monocular video sequences and to recover their 3D motion under changes in their pose and against complex unknown backgrounds.

The remainder of this paper is organized as follows: In Section II, we present details of the data set under investigation and we describe the sequence alignment procedures that divide the motion capture data into cycles. In Section III, we explain our algorithm for functional Principal Component Analysis and we show the principal components in terms of human motion. The application for tracking is described in detail in [1], [2] and is summarized in Section IV. Section V concludes.

II. Human Motion Data

In this work, we develop a modeling procedure for periodic motion sequences. By definition, periodic motion is composed of repetitive "cycles" which constitute a natural unit of statistical modeling and which must be identified in the training data prior to building a model. For example, Figure 2 illustrates one particular cycle of a walking sequence. Frequently, this segmentation is carried out manually in an error-prone and burdensome procedure (see, for example, [3], [4]). In this section, we present alignment algorithms that segment the data automatically. Based on the estimated alignment parameters, the cycles of different motion sequences are then transformed into a common reference domain, and the mean and the principal components of the transformed cycle data are computed as a statistical model.

A. Motion Capture Data

Three-dimensional human motion data can be collected using a variety of commercially available products. Basically, an individual performs an action wearing a specially designed suit with retro-reflective markers located at the joints. The markers are illuminated with infrared LEDs, and their position is tracked with video cameras. From these video images it is easy to reconstruct the the 3D-positions of the markers, allowing for the reconstruction of the observed sequence; see Figure 2 for an example of a walking sequence. For the purpose of our analysis, the marker positions are then translated into 19 body angles to eliminate any global position information and to reduce scaling effects. Hence each "motion sequence" is characterized by a collection of 19 time-series in our case that describe the evolution of relative joint angles over a period of about 500 to 5000 frames. Formally, we let T_i denote the length of the i-th motion sequence and we use $t = 1, \ldots, T_i$ as a

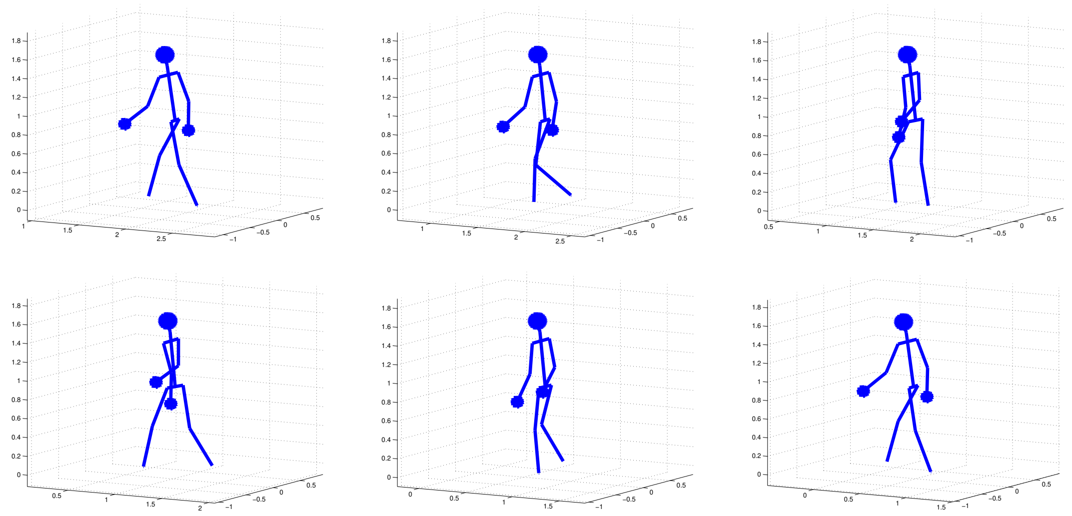

Fig. 2. Computer reconstruction of the body positions observed during a walking sequence.

time index. Similarly, $m = 19$ is the number of angles in each motion sequence and $a = 1, \ldots, m$ indicates a particular angle. The i-th motion sequence can be written as*

$$Z_i(t) \equiv \{z_{i,a}(t) | a = 1, \ldots, m\} \text{ for } t = 0, \ldots, T_i.$$

Associated with each sequence we have the indicator set

$$I_{i,a} \equiv \{t \in \{1, \ldots, T_i\} | z_{i,a}(t) \text{is not missing}\}.$$

That is, $I_{i,a}$ is a set of time indices that labels non-missing informations in $z_{i,a}$ and we have one such set for each sequence i and for each angle a. Missing observations arise frequently in our data set because some markers may be occluded during parts of a motion. The capturing system reports an angle of zero for some of the position coordinates in this case. Typically occlusions last for several frames which prevents

*The data Z_i is not to be confused with the reference coordinate Z_j for limb j in Figure 1.

the imputation of interpolated values using neighboring observations. Below we spend considerable effort to design our algorithms in a manner that is insensitive to this artifact.

Next, we describe a procedure to estimate alignment parameters that segment periodic motion sequences into cycles. The alignment occurs in two steps: For each motion sequence, we first estimate its "cycle length", p, using a "wrapping procedure"; second, we estimate an "offset parameter", o, which describes the relative shift of that sequence using an iterative algorithm. In practice, these parameters vary due to different speeds and different starting positions of the individuals in the motion capture. Based on p and o, the individual motion sequences can be transformed into a common "reference domain" for further analysis.

B. Estimation of the Cycle Length

To estimate the cycle length, we simply try a large number of candidate values for p and we assess the quality of the alignment resulting from p using a simple score function. To illustrate this idea, consider the "wrapping" procedure shown in Figure 3 which uses the left knee angle during walking as an example. Each candidate value p defines a segmentation of the original time series into a sequence of slices that can be represented in the common domain $\{1, \ldots, p\}$. Note that the suitability of p as a choice for the cycle length is easily assessed using this representation: If p is chosen optimally, the individual slices align almost perfectly in the reference

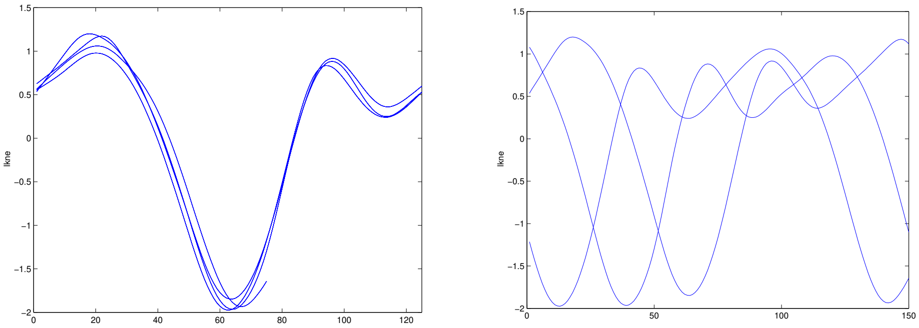

Fig. 3. *Left:* Wrapped representation of the left knee angle during walking using the optimal window size of $p = 125$ ($signal_{lkne} = 2.543483e+02$). *Right:* Wrapped time series using an arbitrary window size, e.g. $p = 150$ ($signal_{lkne} = 4.527145e-01$). There is no obvious alignment in this case.

domain (left). On the other hand, there is significant variation around the mean cycle for suboptimal candidate values (right). Formally, let the "projection index" associated with p be defined according to $\xi_p(t) \equiv t \mod p$, so that ξ_p "folds" the original sequence into the domain $\{1, \ldots, p\}$. For each $k \in \{1, \ldots, p\}$ we will consider all those elements of Z_i that are non-missing and that are projected onto k. The indices of these elements are collected in the sets $I_{i,a}(k) \equiv \{j \in I_{i,a} | k = \xi_p(j)\}$ for $k = 1, \ldots, p$. Also, let $|I_{i,a}(k)|$ be the cardinality of $I_{i,a}(k)$. Then the mean of the observations mapped onto k and the overall mean of sequence i can be written as

$$\begin{aligned}
\bar{z}_{i,a}(k) &\equiv \frac{1}{|I_{i,a}(k)|} \sum_{j \in I_{i,a}(k)} z_{i,a}(j), \\
\bar{Z}_i &\equiv \frac{1}{|I_{i,a}|} \sum_{j \in I_{i,a}} z_{i,a}(j).
\end{aligned}$$

The following magnitudes measure the signal- and the noise-content of the projected sequence:

$$noise_{i,a}(p) \equiv \frac{\sum_{j \in I_{i,a}} (z_{i,a}(j) - \bar{z}_{i,a}(\xi_p(j)))^2}{|I_{i,a}| - p} \quad (1)$$

$$signal_{i,a}(p) \equiv \frac{\sum_{k=1}^{p} (\bar{z}_{i,a}(k) - \bar{Z}_i)^2}{(p-1)p/|I_{i,a}|} \quad (2)$$

Specifically, $noise_{i,a}$ is an estimate of the average square deviation of the aligned cycles from the mean cycle, $\bar{z}_{i,a}$; it can be interpreted as the variation in the data that is not explained by $\bar{z}_{i,a}$. On the other hand, $signal_{i,a}$ is an estimate of the mean variation of the mean cycle, $\bar{z}_{i,a}$, with respect to the overall mean, $\bar{Z}_i$, and measures the signal intensity. Both $noise_{i,a}$ and $signal_{i,a}$ are normalized so as to account properly for the degrees of freedom of the estimates and to produce estimates whose expectation is independent of p. Because it is natural to favor values of p that produce simultaneously high signal- and low noise-contributions, we combine (1) and (2) into the "signal-to-noise ratio":[†]

$$snr_i(p) \equiv \sum_a \frac{signal_{i,a}(p)}{noise_{i,a}(p)}. \quad (3)$$

In our algorithm we try candidate values for p from the set $\{50, 51, \ldots, 250\}$ and choose the maximum with respect to (3) as our estimate of the cycle-length. Note that (3) comprises the accumulated contributions from all angles in our data set. In

[†]Alternative estimation approaches based on spectral analysis cannot easily be applied due to the missing data. Even if there were no missing observations, the statistical efficiency of these estimates might be low due to the relatively small number of frames per period. That is, the conditions of the Sampling Theorem may not be fulfilled so as to guarantee a sufficiently small approximation error.

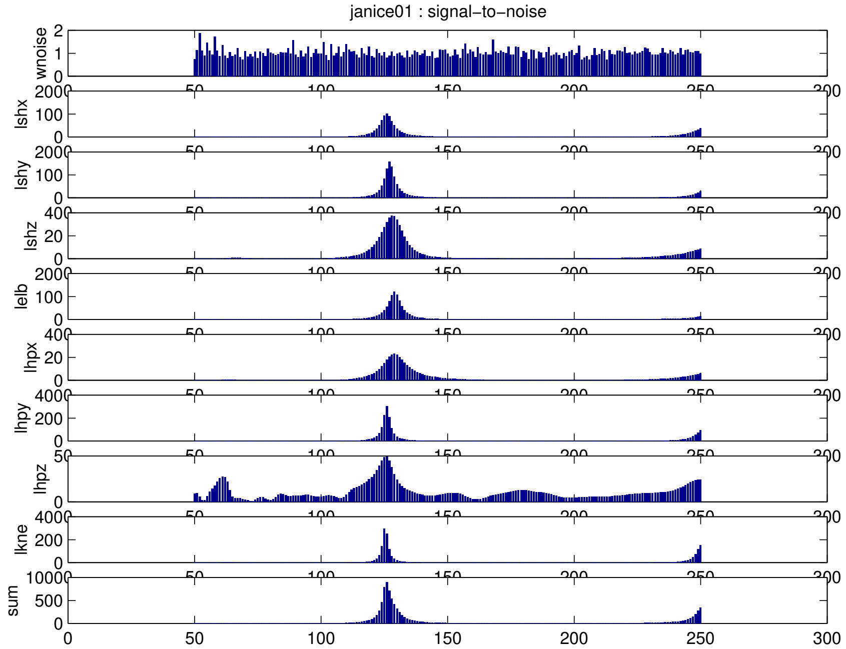

Fig. 4. Signal-to-noise ratio of a representative set of angles as a function of the candidate period length. We show left shoulder (lshx,lshy,lshz), elbow (lelb), hip (lhpx,lhpy,lhpz), and knee angles (lkne). The top series corresponds to a white noise signal. The bottom series shows the accumulated signal-to-noise ratio (3). The minimum overall variation was obtained for $p = 126$, $(snr = 8.986505e+02)$.

Figure 4 we show the individual signal-to-noise ratios for a subset of the angles as well as the accumulated signal-to-noise ratio (3) as functions of p. Note the sharp peak of these values around the optimal cycle length $p = 126$. Note also that the signal-to-noise ratio of the artificially generated white noise series in the first row is approximately constant, warranting the unbiasedness of our approach with respect to changing values of p.

C. Estimation of the Offset Parameter

The folding procedure described above computes an estimate of the optimal cycle length $p(i)$ for each sequence and stores these values in an array of length n. In our second step, we use this array to align multiple sequences in a common domain by rescaling. In detail, we construct offset estimates $o(1), o(2), \ldots, o(n)$ so that the shifted motion sequences minimize the deviation from a common prototype model by analogy to the noise-criterion of the previous paragraph. There are two technical issues complicating this strategy: First, because the motion sequences are typically asynchronous after rescaling using the estimated cycle-lengths, we have to adapt our definitions of the reference signal and of the noise level to a continuous-time framework. Second, we are facing a computational problem. An exhaustive search for the optimal offset-combination requires $O\left(\prod_{i=1}^{n} p(i)\right)$ evaluations of the signal-to-noise criterion, which is clearly infeasible in practice. Instead, we suggest the iterative procedure illustrated in Figure 5 to compute an approximate solution. We initialize the offset values to zero in Step 1, and we define a *reference signal* r_a in Step 2 so as to minimize the deviation with respect to the aligned data. Note that missing data are discarded for estimation. Next, we choose the offsets of all sequences so that they minimize the prediction error with respect to the reference signal (Step 3). By contrast to the exhaustive search, this operation requires only $O\left(\sum_{i=1}^{n} p(i)\right)$ comparisons. Because the solution of the first iteration may well be suboptimal, we construct an improved reference signal using the current offset estimates, and use

1. Initialize offset values. For $i = 1, \ldots, n$, let $o(i) := 0$.

2. From a given function class $\mathcal{R}$, choose the minimum least-squares fit with respect to the aligned data. For $a = 1, \ldots, m$:

$$r_a := \arg\min_{r \in \mathcal{R}} \sum_{i=1}^{n} \sum_{j \in I_{i,a}} \left[z_{i,a}(j) - r\left(\frac{j - o(i)}{p(i)}\right)\right]^2.$$

3. Update the offset parameters. For $i = 1, \ldots, n$:

$$o(i) := \arg\min_{o} \sum_{a} \sum_{j \in I_{i,a}} \left[z_{i,a}(j) - r_a\left(\frac{j - o}{p(i)}\right)\right]^2.$$

4. Stop, if the performance improvement is below 10^{-6}. Otherwise, goto Step 2.

Fig. 5. Iterative algorithm for the computation of the optimal offset parameters.

this signal in turn to improve the offset estimates. Repeating these steps, we obtain an iterative optimization algorithm that is terminated if the improvement falls below a given threshold (Step 4). Since Steps 2 and 3 both decrease the prediction error, it is clear that the algorithm converges.

Figure 6 shows eight sequences of a walking motion, aligned using this procedure. As a function class for the reference signal, $\mathcal{R}$, we chose periodically constrained regression splines in our implementation; i.e. the zeroth, first, and second derivatives of the spline are designed so as to coincide at the boundaries (for details, see Appendix A). Without these constraints, the concatenated reference signal in Figure 6 would be discontinuous at the transitions between cycles.

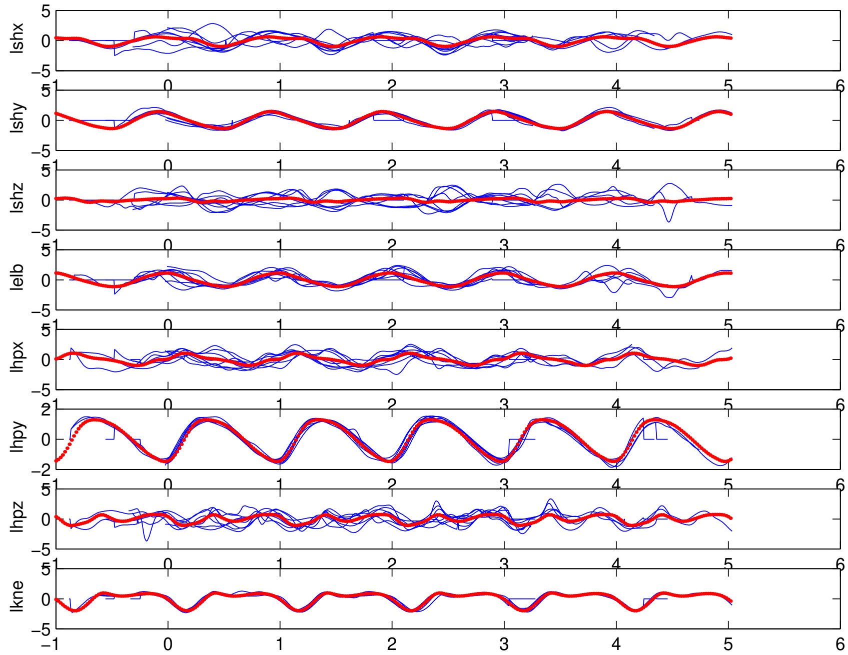

Fig. 6. Aligned representation of eight walking sequences (blue). The red curve denotes repetitions of the periodic regression spline estimate (convergence after five steps).

D. Transformation into Reference Domain

Next, we use this aligned representation to transform the data into a new reference domain. This step is necessary because even though the data are aligned in Figure 6, they are still sampled asynchronously due to the different alignment parameters of each sequence. However, a synchronous matrix representation will be the basis of our principal component algorithm in Section III. Synchronization occurs in two steps: First, we compute function estimates for each cycle of a motion sequence, and second, we resample a new "motion slice" for each cycle in the reference domain $\{0, \ldots, \mathcal{T}\}$. In detail, the sequence $z_{i,a}$ contains $K_i = \lceil \frac{T_i - o(i)}{p(i)} \rceil + 1$ cycles for $o(i) \neq 0$

which are labeled $\tilde{z}_{k,a}$. Here $k = 1, \ldots, K$ ranges over the combined slices from all sequences $i = 1, \ldots, n$ and $K = \sum_{i=1}^{n} K_i$. (We also count the partially filled "ends" on the left and on the right, which are filled up with missing data labels.) $\tilde{z}_{k,a}$ and $\tilde{I}_{k,a}$ denote the k-th slice and the corresponding indicator set, respectively; i.e. $\tilde{I}_{k,a}$ contains the indices of the non-missing elements of $\tilde{z}_{k,a}$. As a function estimate, we compute a (unconstrained) regression spline estimate in the domain $[0, 1]$ for each $k = 1, \ldots, K$ and for each angle:

$$f_{k,a} := \arg\min_{f \in \mathcal{S}} \sum_{j \in \tilde{I}_{k,a}} \left[\tilde{z}_{k,a}(j) - f\left(\frac{j}{p(i)}\right)\right]^2 .$$

Here $\mathcal{S}$ is the class of functions that can be expressed as linear expansions of the basis vectors $\phi_1, \ldots, \phi_K$ defined in Appendix A. This gives a representation of the slices in a continuous domain. Next, we define the reference domain $\{0, \ldots, \mathcal{T}\}$ and we resample the data according to $f_{k,a}$:

$$\bar{z}^{(1)}_{k,a}(j) := f_{k,a}\left(\frac{j}{\mathcal{T}}\right) \text{ for } j = 0, 1, \ldots, \mathcal{T}.$$

The superscript (1) is to distinguish the original data from various approximations in intermediate computing steps described below. For the principal component analysis, it is convenient to rearrange the data in $\bar{z}^{(1)}_{k,a}$ in matrix format. We define the following design matrix for this purpose:

$$X^{(1)} = \begin{pmatrix} \bar{z}^{(1)}_{1,1}(0) & \ldots & \bar{z}^{(1)}_{1,1}(\mathcal{T}) & \ldots & \bar{z}^{(1)}_{1,m}(0) & \ldots & \bar{z}^{(1)}_{1,m}(\mathcal{T}) \\ & & & \ldots & & & \\ \bar{z}^{(1)}_{K,1}(0) & \ldots & \bar{z}^{(1)}_{K,1}(\mathcal{T}) & \ldots & \bar{z}^{(1)}_{K,m}(0) & \ldots & \bar{z}^{(1)}_{K,m}(\mathcal{T}) \end{pmatrix}.$$

That is, each row of $X^{(1)}$ is a concatenation of the slices corresponding to the angles $a = 1, \ldots, m$ and the rows of $X^{(1)}$ correspond to the index $k = 1, \ldots, K$. Below we use the symbols $\bar{z}_{k,a}^{(N)}$ and $X^{(N)}$ to refer to the same data depending on whether we prefer to describe a processing step using vector or matrix format.

III. Principal Component Analysis

Next, we derive a statistical model of the repetitive cycles in Figure 6. For example, we could use the mean cycle as a "prototype" of a motion class to animate a virtual character. For many applications, however, the mean may be insufficient as a motion model. In 3D-animation, for example, it may be desirable to incorporate systematic deviations from the mean to assign personalities to virtual characters. Similarly, in motion tracking we wish to identify motion sequences that deviate from the mean. As a more comprehensive description of the underlying probability distribution, we suggest a new algorithm to estimate the principal components in Figure 7.

In this algorithm the data slices $\bar{z}_{k,a}^{(1)}$ defined above undergo a sequence of processing stages. We use the notation $\bar{z}_{k,a}^{(N)}, N = 1, \ldots, 6$ to denote the intermediate results of this stage-wise computation, and we interpret $\bar{z}_{k,a}^{(1)}$ as the first element of this sequence. Similarly, $X^{(N)}$ is the design matrix at stage N which is composed of the data in $\bar{z}_{k,a}^{(N)}, k = 1, \ldots, K, a = 1, \ldots, m$. Our algorithm addresses several difficulties: First, because of the missing data in $X^{(1)}$, we cannot simply use a standard Singular Value Decomposition (SVD) to obtain the principal components. As an alternative approach, we use an iterative approximation scheme suggested recently by Hastie et

1. Compute the row-mean μ of $X^{(1)}$, and let $X^{(2)} := X^{(1)} - 1^T\mu$. Here 1 is a vector of ones. Initialize missing values in $X^{(2)}$ using zeros.

2. For each of the slices $\bar{z}_{k,a}^{(2)}$ in $X^{(2)}$, compute the first 20 Fourier coefficients and store the result in a new matrix, $X^{(3)}$.

3. Compute the Singular Value Decomposition of $X^{(3)}$:

$$X^{(3)} = USV^T.$$

4. Reconstruct the design matrix, using the rank q approximation to S:

$$X^{(4)} = US^qV^T. \tag{4}$$

5. Apply the Inverse Fourier Transform to obtain $X^{(5)}$.

6. Evaluate the reconstruction error $||X^{(2)} - X^{(5)}||$. Update the imputation of the missing values:

$$\bar{z}_{k,a}^{(2)}(j) = \bar{z}_{k,a}^{(5)}(j) \text{ for all } j \in \bar{I}_{k,a}.$$

7. If the error reduction is above 10^{-6} goto Step 2.

8. Return $X^{(6)} = X^{(5)} + 1^T\mu$.

Fig. 7. Functional PCA algorithm with data imputation and enforcement of smoothness constraints.

al. in the context of analyzing Gene Expression Arrays [5]. In detail, we alternate between an SVD step (2 through 5) and a data imputation step (Step 6), where each update is designed so as to decrease the matrix distance between the original data and their reconstruction. As an additional complication, we cannot compute the SVD directly because the principal components obtained in this manner could be non-periodic or even discontinuous. A pragmatic approach is to project the data onto a smooth and periodic basis prior to carrying out the SVD (Step 2). We choose a Fourier basis for this purpose, and we truncate the high-frequency components by keeping only the 20 leading coefficients. This gives a new coefficient matrix $X^{(3)}$. Next, we compute the optimal rank-q approximation to $X^{(3)}$ using a complex SVD in Steps 3 and 4, and we reconstruct the signal in the original domain in Step 5. In Step 6 we use the reconstructed values as improved estimates for the missing data in $X^{(2)}$, and then we repeat Steps 2 through 5 using these improved estimates. This iterative process is continued until the performance improvement falls below a given threshold.

Theorem 1: The Functional Principal Component Algorithm in Figure 7 monotonically minimizes the matrix distance between the normalized design matrix and its reconstruction, $||X^{(2)} - X^{(5)}||$.

Proof: It is clear that the data imputation step 6 decreases the reconstruction error because the differences between the imputations of missing values are set to zero. Steps 2 through 5 project $X^{(2)}$ onto the linear subspace spanned by the low-frequency Fourier-basis. To see this, note that the Steps 2 and 5 can be written alternatively as matrix multiplications

$$X^{(3)} = X^{(2)}B \quad \text{and} \quad X^{(5)} = X^{(4)}B^T = US^qV^TB^T.$$

Here $\tilde{B} = (B, B_\perp)$ describes the angle-wise Fourier transform and B is the submatrix of $\tilde{B}$ that corresponds to the first 20 coefficients. $B_\perp$ determines the remaining coefficients and is orthogonal to B, i.e. $B^T B_\perp = 0$. Because the Fourier basis is orthonormal, B and $B_\perp$ satisfy the conditions $B^T B = I$ and $B_\perp^T B_\perp = I$, respectively. The reconstruction error can be decomposed as follows:

$$\begin{aligned} ||X^{(5)} - X^{(2)}||^2 &= ||(X^{(5)} - X^{(3)})B||^2 + ||(X^{(5)} - X^{(2)})B_\perp||^2 \\ &= ||US^qV^TB^TB - X^{(2)}B||^2 + ||US^qV^TB^TB_\perp - X^{(2)}B_\perp||^2 \\ &= ||US^qV^T - X^{(2)}||^2 + ||X^{(2)}B_\perp||^2. \end{aligned} \tag{5}$$

The first term in (5) is the rank-q reconstruction error of $X^{(2)}$ which is minimized by the values of U, S, and V computed in Step (4). Hence $X^{(5)}$ is the optimal rank-q approximation of $X^{(2)}$ within the class of matrices whose row-space is spanned by the rows of B. In other words, $X^{(5)}$ is a orthogonal projection of $X^{(2)}$ with respect to $||\cdot||$ and the second term in (5), which is independent of U, S, and V, corresponds to the residuals of this projection. Because $X^{(5)}$ minimizes $||X^{(5)} - X^{(2)}||$, it is superior to the $X^{(5)}$ from the previous iteration in particular. Hence both the data imputation and the SVD steps reduce $||X^{(5)} - X^{(2)}||$ which gives the desired result.

■

In Figure 8 we show the mean and the four principal components for various body angles estimated from the walking data. Because a common mean was computed for different individuals in this figure, the principal components can be interpreted as the

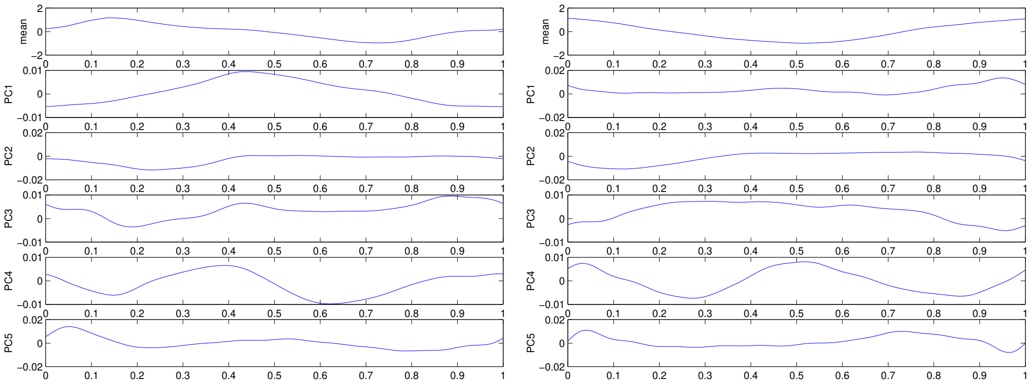

Fig. 8. *Left:* Mean and the four leading principal components of the left hip-angle during walking, estimated from eight sequences. The singular values corresponding to these principal components are 891.9855, 410.4835, 362.1642, 290.5781, 255.2503. *Right:* The same data for the left elbow. The singular values of these data are 631.8444, 513.3287, 430.0021, 356.7820, 246.7720.

main sources of variation *between* different individuals. In Figure 9 we show the same

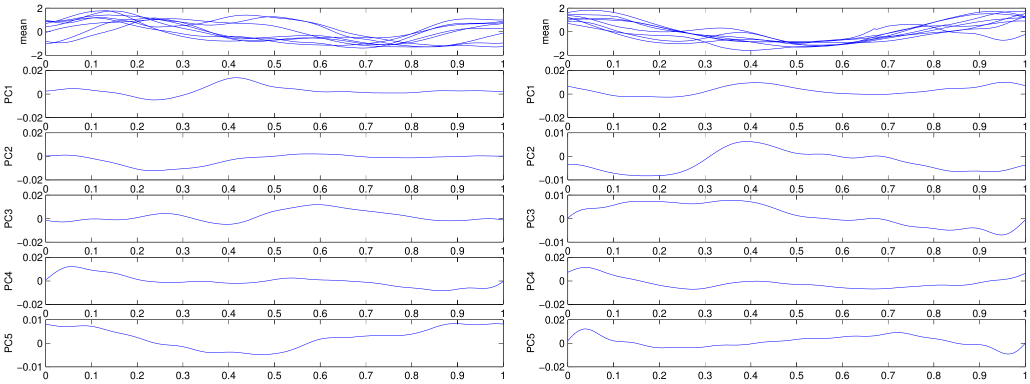

Fig. 9. *Left:* Mean and the four leading principal components of the left hip-angle during walking, estimated from eight sequences. The singular values corresponding to these principal components are 497.4937, 465.9257, 365.1190, 273.8710, 211.8135. *Right:* The same data for the left elbow. The singular values of these data are 698.7418, 542.4724, 390.4070, 285.3837, 206.1909.

principal components using different means for different individuals. Hence, the principal components characterize variations *within* different realizations by one and the same individual in this case. An intuitive way to interpret these data is by using animation. Here we add a relative large multiple of each component to the mean cycle and visualize the resulting motion sequence. Another interesting experiment is to artificially generate new motion sequences by using different random combinations of the principal components. Corresponding movies are available on the World Wide Web at `http://www-stat.stanford.edu/~ormoneit/vision/3Dtracking.html`.

IV. Motion Tracking in Video Sequences

In this section, we describe an application of the principal component model for the tracking of human motion in monocular video sequences. In our approach to tracking, the goal is to estimate the joint angles of the body and its 3D pose given a sequence of 2D image measurements, $\mathbf{J}_t \equiv \{J_t, \ldots, J_1\}$. Of course, 2D image data is insufficient in general to infer 3D information exactly. As an approximation, we interpret the human 3D poses as unobserved variables, ϕ_t, in a state-space model and we treat the image data J_t as observations. Here the principal components serve to define a low-dimensional representation of the human 3D poses which is particularly important to reduce the computational burden and to incorporate prior information. Such additional information regarding the type of motion that can be expected in a video sequence is crucial to simplify the complex tracking task.

A decisive aspect of the specification of the state-space model is the observation equation that incorporates a complex generative model of image appearance. For example, the observation equation includes features such as the camera geometry, the articulated human body model of Figure 1, texture- and color-maps, as well as a projection and an occlusion model. Because our focus in this work is not on computer vision but on the modeling aspects of human motion tracking we shall not discuss these issues here. Instead, we refer the reader interested in details to [1], [2], as well as to the references mentioned therein.

Formally, let $\theta(t) \equiv (\theta_a(t) | a = 1, \ldots, m)$ be a random vector summarizing the relative joint angles at time t; in other words, the motion sequence, $Z_i(t)$, at time t is now interpreted as the i-th realization of $\theta(t)$. Under the modeling assumptions of the SVD in Figure 7, $\theta(t)$ can be written in the form

$$\theta(t) = \tilde{\mu}(\psi_t) + \sum_{k=1}^{q} c_{t,k} v_k(\psi_t),$$

where v_k is the Fourier inverse of the k-th column of V in (4), rearranged as a $\mathcal{T} \times m$-matrix; similarly, $\tilde{\mu}$ denotes the rearranged mean vector μ in Figure 7. $v_k(\psi)$ is the ψ-th column of v_k, and the $c_{t,k}$ are time-varying coefficients. $\psi_t \in \{0, \mathcal{T} - 1\}$ maps absolute time onto relative cycle positions or phases, and ρ_t denotes the speed of the motion such that $\psi_{t+1} = (\psi_t + \rho_t) \bmod \mathcal{T}$. Also, let $\mathbf{c}_t = (c_{t,1}, \ldots, c_{t,q})$ denote a vector of the q linear coefficients and let $\boldsymbol{\tau}_t^g$ and $\boldsymbol{\theta}_t^g$ represent the global 3D translation and rotation of the torso. Then body positions are characterized by the state-vector

$$\boldsymbol{\phi}_t = (\mathbf{c}_t, \psi_t, \rho_t, \boldsymbol{\tau}_t^g, \boldsymbol{\theta}_t^g)^T.$$

The dynamics of a state-space model are described by the densities:

$$p(\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t-1}) \quad \text{“Transition Model”}$$

$$p(J_t|\boldsymbol{\phi}_t) \quad \text{“Observation Model”}.$$

In other words, the “transition model” or “transition density” $p(\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t-1})$ characterizes the random change in $\boldsymbol{\phi}_t$ from time $t-1$ to t, and the “observation model” $p(J_t|\boldsymbol{\phi}_t)$ characterizes the generation of images. Details of these densities are described in [1], [2]. In tracking, interest at time t focuses on the conditional distribution $p(\boldsymbol{\phi}_t|\mathbf{J}_t)$. This is because we would like to recover the probability distribution over the 3D body poses given the sequence of previous images. One possibility is to compute $p(\boldsymbol{\phi}_t|\mathbf{J}_t)$ recursively using the *prediction* and *filtering* equations:

$$p(\boldsymbol{\phi}_t|\mathbf{J}_{t-1}) = \int p(\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t-1})p(\boldsymbol{\phi}_{t-1}|\mathbf{J}_{t-1})d\boldsymbol{\phi}_{t-1} \tag{6}$$

$$p(\boldsymbol{\phi}_t|\mathbf{J}_t) \propto p(J_t|\boldsymbol{\phi}_t)p(\boldsymbol{\phi}_t|\mathbf{J}_{t-1}). \tag{7}$$

(6) and (7) can be evaluated in closed form provided the transition and the observation models are linear with Gaussian noise. Unfortunately, the observation equation resulting from the generative image model is highly non-linear so that we have to resort to approximate inference for filtering. We use a so-called “particle filter” for this purpose where $p(J_t|\boldsymbol{\phi}_t)$ is represented as a weighted set of samples that are

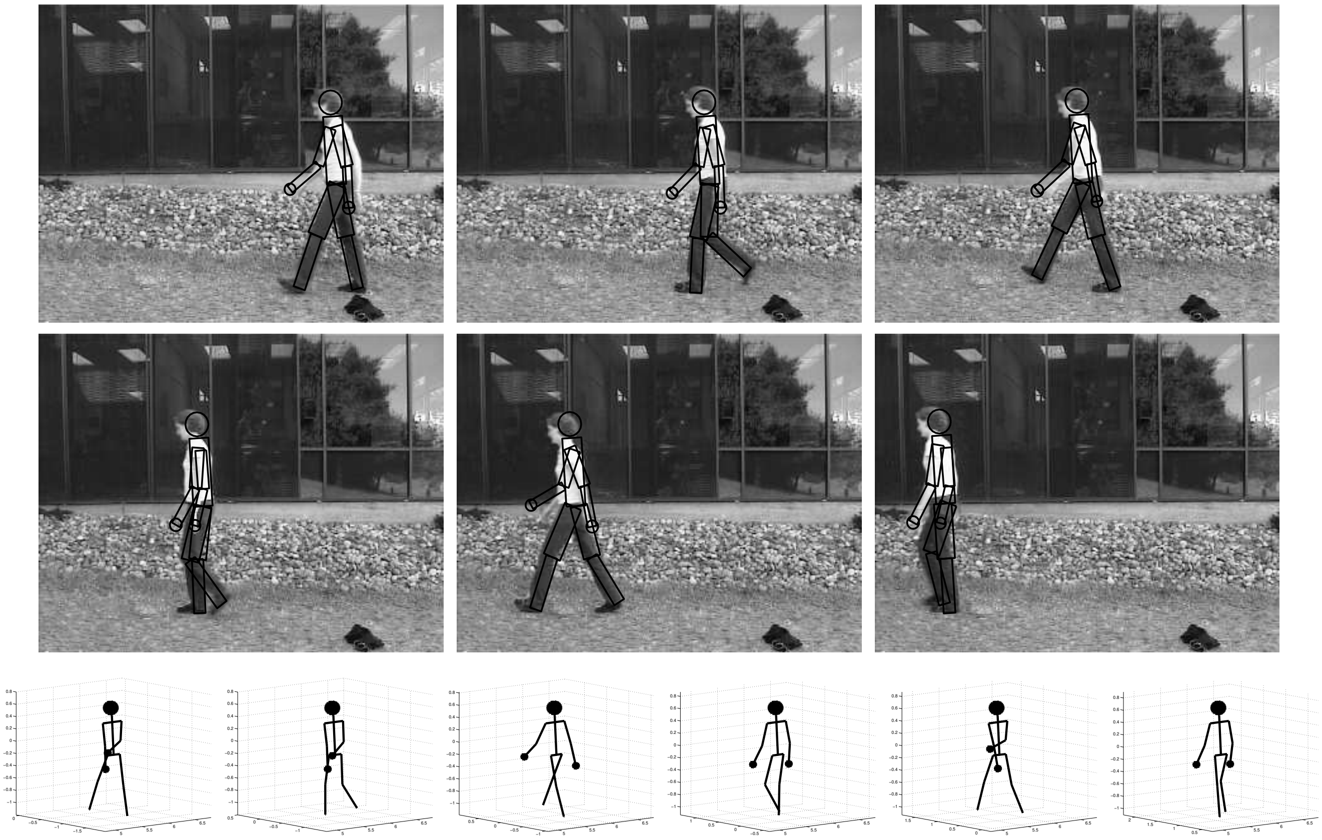

Fig. 10. Tracking of person walking, 10000 samples. The two upper rows show frames 0, 10, 20, 30, 40, 50 in the sequence with the projection of the expected model configuration overlayed. The lower row shows the expected 3D configuration in the same frames.

propagated in time (e.g. [6], [7]).

To illustrate the method we show an example of tracking a walking person in a cluttered scene. On an Ultra 1 Sparc station the C++ implementation ran at a rate of approximately 1 frame per minute. To visualize the posterior distribution we display the projection of the 3D model corresponding to the expected value of the model parameters. All parameters were initialized with a Gaussian prior at time $t = 0$.

Figure 10 shows the tracking results for frames 0 to 50 of a walking sequence. Note that the legs of the model are better aligned with the image data than the arms. This is probably due to the fact that the arms are more often occluded by the torso, and thus more prior driven than the legs. In parts of the cycle where large occlusion occurs (frame 30) the model has little image information, and starts to drift off the person. However, it recovers when a larger part of the body is visible (frame 40). Again, video animations of these results can be found online at `http://www-stat.stanford.edu/~ormoneit/vision/3Dtracking.html`.

V. Conclusion

This paper describes a fully automated method for learning periodic human motions from training data. Statistical methods are presented for detecting the length of the periods in the data, segmenting it into cycles, and optimally aligning the cycles. We also presented a novel principal component analysis technique for modeling the motion curves. The method copes with missing data and enforces smoothness between the beginning and ending of a motion cycle. The principal components serve to define a state-space model for the tracking of human motion in video sequences. We have demonstrated results for tracking a person in a cluttered image sequence.

Acknowledgments

The work of Dirk Ormoneit was partly supported by a grant of the Deutsche Forschungsgemeinschaft (DFG). We are grateful to Michael Gleicher for generously providing the 3D motion-capture data used in our experiments and to Hedvig Sidenbladh for sharing the results of her computer simulations. We thank Manolis Kamvysselis and David Fleet for many interesting discussions about human motion.

Appendix

I. Periodic Regression Splines

We consider a regression spline estimate in the domain $[0, 1]$ with fixed knots $\check{x}_k \in \{0.1, 0.2, \ldots, 0.9\}$. The regression spline, $f(x)$, is a linear expansion of the basis functions $\phi_k(x)$, $k = 0, \ldots, K$ (for details, see Section 3.2.5 in [3]):

$$\phi_0(x) \equiv 1, \;\; \phi_1(x) \equiv x, \;\; \phi_2(x) \equiv x^2, \;\; \phi_3(x) \equiv x^3, \;\; \phi_{k+3}(x) \equiv (x - \check{x}_k)^3_+.$$

These basis functions are summarized in the function $\Phi(x) \equiv (\phi_1(x), \ldots, \phi_K(x))^T$ so that $f(x) = \Phi(x)^T \beta$ using a $K+1$ vector of coefficients β. Our objective is to enforce boundary constraints of the form $f^{(i)}(0) = f^{(i)}(1)$ for $i = 0, 1, 2$ where $f^{(i)}(x)$ is the i-th derivative of $f(x)$ at x. We formulate these constraints using matrix notation as

$$C^T \beta = 0$$

where

$$C^T \equiv \begin{pmatrix} 0 & \phi_1(0) - \phi_1(1) & \dots & \phi_K(0) - \phi_K(1) \\ 0 & \phi_1^{(1)}(0) - \phi_1^{(1)}(1) & \dots & \phi_K^{(1)}(0) - \phi_K^{(1)}(1) \\ 0 & \phi_1^{(2)}(0) - \phi_1^{(2)}(1) & \dots & \phi_K^{(2)}(0) - \phi_K^{(2)}(1) \end{pmatrix}.$$

Note that C is a $(K+1) \times 3$ matrix, and consider the Singular Value Decomposition

$$C = USV^T = (U_1 \quad U_2) \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}.$$

Here S_1 corresponds to the part of S with non-zero diagonal entries. Next, observe that $C^T\beta = V_1 D_1 U_1^T \beta$ equals zero if and only if $U_1^T\beta = 0$. Using $UU^T = I$, we can rewrite $f(x)$ as follows:

$$\begin{aligned} f(x) &= \Phi(x)^T\beta \\ &= \Phi(x)^T UU^T\beta \\ &= \left(\Phi(x)^T U_1 \quad \Phi(x)^T U_2\right) \begin{pmatrix} U_1^T\beta \\ U_2^T\beta \end{pmatrix} \\ &= \Phi(x)^T U_2 U_2^T \beta \\ &= \tilde{\Phi}(x)^T \tilde{\beta}. \end{aligned}$$

Hence $f(x)$ is also the solution of an unconstrained regression using the transformed basis vector $\tilde{\Phi}(x) = \Phi(x)^T U_2$. The computation of the coefficient vector $\tilde{\beta} = U_2^T\beta$ is straightforward using this transformation.

References

[1] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3D human figures using 2D image motion," in *EECV, Dublin, Ireland*, 2000.

[2] D. Ormoneit, H. Sidenbladh, M. J. Black, and T. Hastie, "Learning and tracking human motion using functional analysis," in *IEEE Workshop on Human Modeling, Analysis and Synthesis*, 2000.

[3] J. O. Ramsey and B. W. Silverman, *Functional Data Analysis*, Springer-Verlag, 1997.

[4] Y. Yacoob and M. Black, "Parameterized modeling and recognition of activities in temporal surfaces," in *CVIU*, 1999, vol. 73, pp. 232–247.

[5] T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, O. Alter, D. Botstein, and P. Brown, "Imputing missing data for gene expression arrays," 2000, Working Paper, Department of Statistics, Stanford University.

[6] J. Liu and R. Chen, "Sequential monte carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, pp. 1032–1044, 1998.

[7] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *ECCV*, 1996, pp. 343–356.