Temporal Motion Models for Monocular and Multiview 3–D Human Body Tracking ⋆

Raquel Urtasun [a] David J. Fleet [b] Pascal Fua [a]

[a] *Computer Vision Laboratory*
Swiss Federal Institute of Technology (EPFL)
1015 Lausanne, Switzerland

[b] *Department of Computer Science*
University of Toronto
M5S 3H5, Canada

Abstract

We explore an approach to 3D people tracking with learned motion models and deterministic optimization. The tracking problem is formulated as the minimization of a differentiable criterion whose differential structure is rich enough for optimization to be accomplished via hill-climbing. This avoids the computational expense of Monte Carlo methods, while yielding good results under challenging conditions. To demonstrate the generality of the approach we show that we can learn and track cyclic motions such as walking and running, as well as acyclic motions such as a golf swing. We also show results from both monocular and multi-camera tracking. Finally, we provide results with a motion model learned from multiple activities, and show how this models might be used for recognition.

Key words: Tracking, Motion Models, Optimization
PACS:

1 Introduction

Prior models of pose and motion play a central role in 3D monocular people tracking, mitigating problems caused by ambiguities, occlusions, and image measurement noise. While powerful models of 3D human *pose* are emerging, there has been comparatively little work on *motion* models [1–4]. Most state-of-the-art approaches rely on simple Markov models that do not capture the complexities of

⋆ This work was supported in part by the Swiss National Science Foundation, NSERC Canada and the Canadian Institute of Advanced Research

human dynamics. This often produces a more challenging inference problem for which Monte Carlo techniques (e.g., particle filters) are often used to cope with ambiguities and local minima [5–9]. Most such methods suffer computationally as the number of degrees of freedom in the model increases.

In this paper, we use activity-specific motion models to help overcome this problem. We show that, while complex non-linear methods are required to learn pose models, one can use simple algorithms such as PCA to learn effective motion models, both for cyclic motions such as walking and running, and acyclic motions such as a golf swing. With such motion models we formulate and solve the tracking problem in terms of continuous objective functions whose differential structure is rich enough to take advantage of standard optimization methods. This is significant because the computational requirements of these methods are typically less than those of Monte Carlo methods. This is demonstrated here with two tracking formulations, one for monocular people tracking, and one for multiview people tracking. Finally, with these subspace motion models we also show that one can perform motion-based recognition of individuals and activities.

2 Related Work

Modeling and tracking the 3D human body from video is of great interest, as attested by recent surveys [10,11], yet existing approaches remain brittle. The causes of the main problems include joint reflection ambiguities, occlusion, cluttered backgrounds, non-rigidity of tissue and clothing, complex and rapid motions, and poor image resolution. People tracking is comparatively simpler if multiple calibrated cameras can be used simultaneously. Techniques such as space carving [12,13], 3D voxel extraction from silhouettes [14], fitting to silhouette and stereo data [15–17], and skeleton-based techniques [18,19] have been used with some success. If camera motion and background scenes are controlled, so that body silhousttes are easy to extract, these techniques can be very effective. Nevertheless, in natural scenes, with monocular video and cluttered backgrounds with significant depth variation, the problem remains very challenging.

Recent approaches to people tracking can be viewed in terms of those that *detect* and those that *track*. Detection, involving pose recognition from individual frames, has become increasingly popular in recent research [20–24] but requires large numbers of training poses to be effective. Tracking involves pose inference at one time instant given state information (e.g., pose) from previous time instants. Tracking often fails as errors accumulate through time, producing poor predictions and hence divergence. Often this can be mitigated with the use of sophisticated statistical techniques for a more effective search [7,5,25,6,9], or by using strong prior motion models [26,27,8].

Detection and tracking are complementary in many respects. Tracking takes advantage of temporal continuity and the smoothness of human motions to accumulate information through time, while detection techniques are likely to be useful for initialization of tracking and search. With suitable dynamical models, tracking has the added advantage of providing parameter estimates that may be directly relevant for subsequent recognition tasks with applications to sport training, physiotherapy or clinical diagnostics. In this paper we present a tracking approach in which simple detection techniques are used to find key postures and thereby provide rough initialization for tracking.

Dynamical models may be generic or activity specific. Many researchers adopt generic models that encourage smoothness while obeying kinematic joint limits [5,28,29,9] . Such models are often expressed in terms of first- or second-order Markov models. Activity-specific models more strongly constrain 3D tracking and help resolve potential ambiguities, but at the cost of having to infer the class of motion, and to learn the models.

The most common approach to learning activity-specific models of motion or pose has been to use optical motion capture data from one or more people performing one or more activities. Given the high-dimensionality of the data it is natural to look for low-dimensional embeddings of the data (e.g., [30]). To learn pose models a key problem concerns the highly nonlinear space of human *poses*. Accordingly, methods for nonlinear dimensionality reduction have been popular [21,31–34].

Instead of modeling the *pose* space, one might directly model the space of human *motions*. Linear subspace models have been used to model human motion, from which realistic computer animations have been produced [35–38]. Subspace models learned from multiple people performing the same activity have been used successfully for 3D people tracking [27,8,39]. For the restricted class of cyclic motions, Ormoneit et al. [27] developed an automated procedure for aligning training data as a precursor to PCA. Troje [40] considers a related class of subspace models for walking motions in which the temporal variations in pose is expressed in terms of sinusoidal basis functions. He finds that three harmonics are sufficient for reliable gender classification from optical motion capture data.

3 Motion Models

This paper extends the use of linear subspace methods for 3D people tracking. In this section we describe the protocol we use to learn cyclic and acyclic motions, and then discuss the important properties of the models. We show how they tend to cluster similar motions, and that the linear embedding tends to produce convex models. These properties are important for the generalization to motions outside of the training set, to facilitate tracking with continuous optimization, and for motion-

based recognition.

We represent the human body as the set of volumetric primitives attached to an articulated 3–D skeleton, like those depicted in Figs. 12 and 14. A *pose* is given by the position and orientation of its root node, defined at the sacroiliac, and a set of joint angles. More formally, let D denote the number of joint angles in the skeletal model. A pose at time t is then given by a vector of joint angles, denoted $\psi_t = [\theta_1, \cdots, \theta_D]^T$, along with the global position and orientation of the root, denoted $\mathbf{g}_t \in \mathcal{R}^6$.

A *motion* can be viewed as a time-varying pose. While pose varies continuously with time, we assume a discrete representation in which pose is sampled at N distinct time instants. In this way, a motion is just a sequence of N discrete poses:

$$\Psi = [\psi_1^T, \cdots, \psi_N^T]^T \in \mathcal{R}^{D\,N} \quad , \tag{1}$$

$$\mathbf{G} = [\mathbf{g}_1^T, \cdots, \mathbf{g}_N^T]^T \in \mathcal{R}^{6\,N} \quad , \tag{2}$$

Naturally, we assume that the temporal sampling rate is sufficicently high that we can interpolate the continuous pose signal.

A given motion can occur at different speeds. In order to acheive some speed independence we encode the motion for a canonical speed, from which time warping can be used to create other speeds. For the canonical motion representation we let the pose vary as a function of a phase parameter μ that is defined to be 0 at the begining of the motion and 1 at the end of the motion. For periodic motions defined on a circle, like walking, the phase is periodic. The canonical motion is then represented with a sequence of N poses, indexed by the phase of the motion. For frame $n \in [1,\ N]$, the discrete phase $\mu_n \in [0, 1]$ is simply

$$\mu_n = \frac{n-1}{N-1}\,. \tag{3}$$

3.1 PCA Motion Model

We learn motion models from optical motion capture data comprising one or more people performing the same activity several times. Because different people perform the same activity with some variability in speed, we first dynamically time-warp and re-sample each training sample. This produces training motions with the same number of samples, and with similar poses aligned (to obtain the canonical reference frame). To this end, we first manually identify a small number of key postures specific to each motion type. We then linearly time warp the motions so that the key postures are temporally aligned. The resulting motions are then re-sampled at regular time intervals using quaternion spherical interpolation [41] to produce the training poses $\{\psi_j\}_{j=1}^N$.

Given a training set of M such motions, denoted, $\{\Psi_j\}_{j=1}^{M}$, we use Principal Component Analysis to find a low-dimensional basis with which we can effectively model the motion. In particular, the model approximates motions in the training set with a linear combination of the mean motion Θ_0 and a set of *eigen-motions* $\{\Theta_i\}_{i=1}^{m}$:

$$\Psi \approx \Theta_0 + \sum_{i=1}^{m} \alpha_i \Theta_i \quad . \tag{4}$$

The scalar coefficients, $\{\alpha_i\}$, characterize the motion, and $m \leq M$ controls the fraction of the total variance of the training data that is captured by the subspace, denoted by $Q(m)$:

$$Q(m) = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{M} \lambda_i} \, , \tag{5}$$

where λ_i are the eigenvalues of the data covariance matrix, ordered such that $\lambda_i \geq \lambda_{i+1}$. In what follows we typically choose m so that $Q(m) > 0.9$.

A pose is then defined as a function of the scalar coefficients, $\{\alpha_i\}$, and a phase value, μ, i.e.

$$\psi(\mu, \alpha_1, \cdots, \alpha_m) \approx \Theta_0(\mu) + \sum_{i=1}^{m} \alpha_i \Theta_i(\mu) \quad . \tag{6}$$

Note than now $\Theta_i(\mu)$ are *eigen-poses*, and $\Theta_0(\mu)$ is the mean pose for that particular phase.

3.2 Cyclic motions

We first consider models for walking and running. We used a Vicon^{tm} optical motion capture system to measure the motions of two men and two women on a treadmill:

- walking at 9 speeds ranging from 3 to 7 km/h, by increments of 0.5 km/h, for a total of 144 motions;
- running at 7 speeds ranging from 6 to 12 km/h, by increments of 1.0 km/h, for a total of 112 motions.

The body model had $D = 84$ degrees of freedom. While one might also wish to include global translational or orientational velocities in the training data, these were not available with the treadmill data, so we restricted ourselves to temporal models of the joint angles. The start and end of each gait cycle were manually identified. The data were thereby broken into individual cycles, and normalized so that each gait cycle was represented with $N = 33$ pose samples. Four cycles of walking and running at each speed were used to capture the natural variability of motion from one gait cycle to the next for each person.

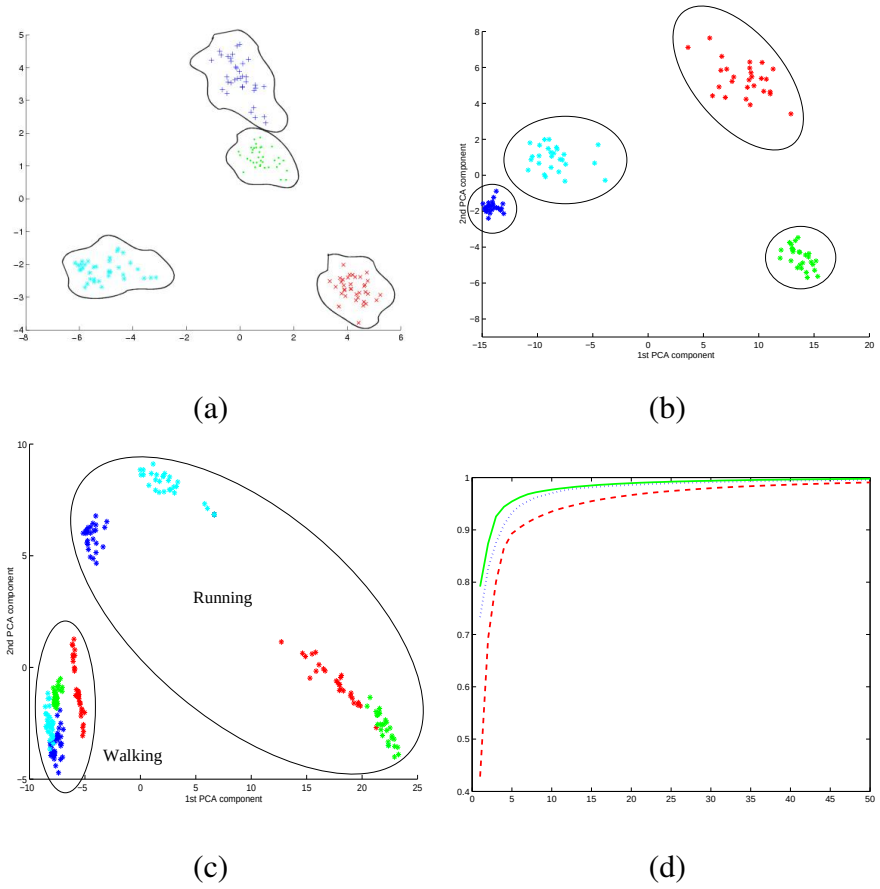


Fig. 1. Motion models. First two PCA components for (a) 4 different captures of 4 subjects walking at speeds varying from 3 to 7km/h, (b) the same subjects at speeds ranging from 6 to 12km/h, (c) the multi-activity database composed of the walking and running motions together. The data corresponding to different subjects is shown in different styles. The solid lines separating clusters have been drawn manually for visualization purposes. (d) Percentage of the database that can be generated with a given number of eigenvectors for the walking (dashed red), running(solid green) and the multi-activity databases(dotted blue).

In what follows we learn a motion model for walking and one for running, as well as multi-activity model for the combined walking and running data. In Fig. 1(d) we display $Q(m)$ in (5) as a function of the number of eigen-motions for the walking, running and the combined datasets. We find that in all three cases $m = 5$ eigen-motions out of a possible 144 for walking, 112 for running and 256 for the multi-activity data, capture more than 90% of the total variance. In the experiments below we show that these motion models are sufficient to generalize to styles that were not captured in the training data, while eliminating the noise present in the less significant principal directions.

The first five walking eigen-motions, Θ_i, for the upper and lower leg rotations in the sagittal plane are depicted by Fig. 2 as a function of the gait phase μ_t. One can see that they are smooth and therefore easily interpolated and differentiated numerically by finite differences. Fig. 3 illustrates the individual contributions of

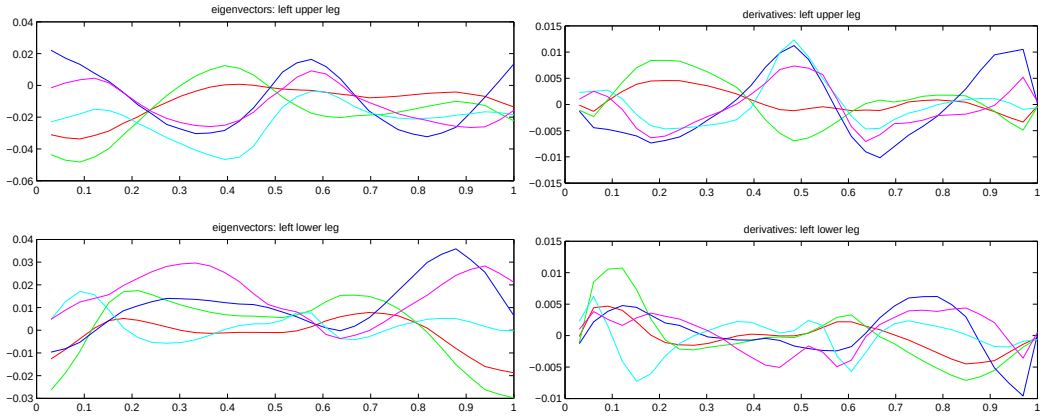


Fig. 2. Walking motion eigenvectors (left) and their derivatives (right). The first one appears in red, green the second, blue the third, cian the fourth and magenta for the last one. Eigenvectors and their derivatives, $\frac{\partial \Theta_{ij}}{\partial \mu_t}$, of the flexion-extension in the sagittal plane of the upper leg on the top and flexion-extension in the sagittal plane of the lower leg on the right.

the first five eigen-motions. The first row shows the mean motion alone. In each subsequent row we show a linear combination of the mean motion and the i^{th} eigen-motion, for $i = 1...5$. Each row therefore illustrates the influence of a different eigen-motion. While one cannot expect the individual eigen-motions to have any particular semantic meaning, their behaviour provides some intuitions about the nature of the underlying model.

3.3 Golf Swing

We use the same approach to learn the swing of a golf club. Toward this end, we used the $M = 9$ golf swings of the CMU database [42]. The corresponding body model has $D = 72$ degrees of freedom. We identified the 4 key postures depicted in Fig. 4, and piecewise linearly time-warped the swings so that the same key postures are temporally aligned. We then sampled the warped motions to obtain motions vectors with $N = 200$ poses. The sampling rate here is higher than the one used for walking and running since a golf swing contains fast speeds and large accelerations. Given the small number of available training motions we only used $m = 4$ coefficients, capturing more than 90% of the total variance.

3.4 Motion Clustering

Troje [40] showed that with effective motion models one can perform interesting motion-based recognition. In particular one can classify gender and other individual attributes including emotional states. In this context it is of interest to note that the subspace motion models learned here exhibit good inter-subject and inter-activity

Fig. 3. The top row shows equispaced poses of the mean walk. The next 5 rows illustrate the influence of the first 5 egien-motions. The second row shows a linear combination of the mean walk and the first eigen-motion, $\Theta_0 + 0.7\Theta_1$. Similarly, the third row depicts $\Theta_0 + 0.7\Theta_2$ to show the influence of the second eigen-motion, and so on for the remaining 3 rows.

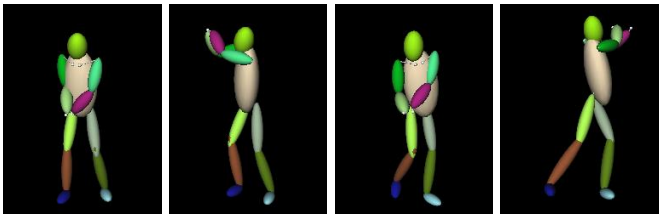
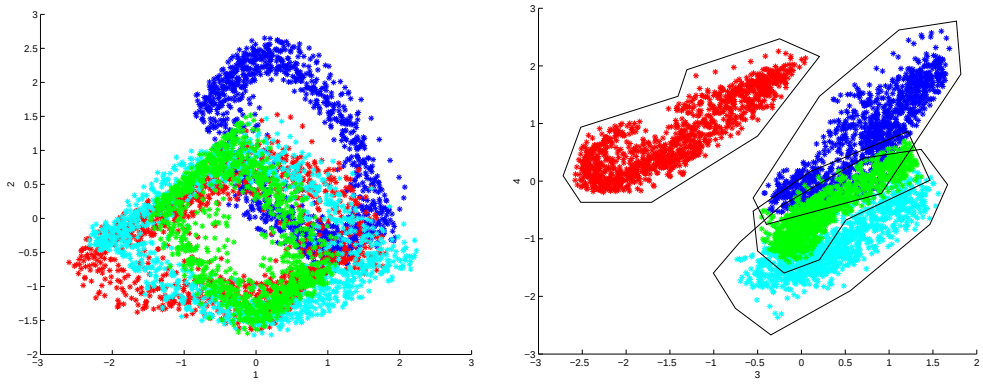


Fig. 4. Key postures for the golf swing motion capture database that are used to align the training data: Beginning of upswing, end of upswing, ball hit, and end of downswing. The body model is represented as volumetric primitives attached to an articulated skeleton.



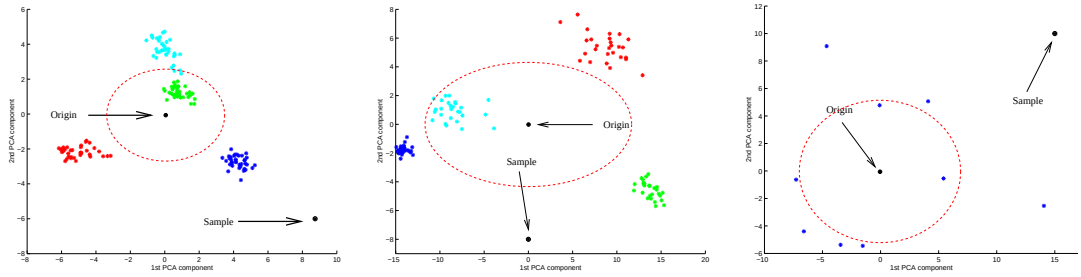
(a) (b)

Fig. 5. Poor clustering in a pose subspace. The solid lines that delimited clusters have been manually done for visualization purposes. (a) Projection of training poses onto the first two eigen-directions of the pose subspace. (b) Projection of training poses onto the third and fourth eigen-directions of the pose subspace. While in the motion motion there is strong inter-subject separation, with the pose model in this figure, there is no inter-subject separation.

separation, suggesting that these models may be useful for recognition. For example, Fig. 1a shows the walking training motions, at all speeds, projected onto the first two eigen-motions of the walking model. Similarly, Fig. 1b shows the running motions, at all speeds, projected onto the first two eigen-motions of the running model. The closed curves in these figures were drawn manually to help illustrate the large inter-subject separation. One can see that the intra-subject variation in both models is much smaller than the inter-subject variation.

The motion model learned from the combination of walking and running training data shows large inter-activity separation. Fig. 1c shows the projection of the training data onto the first two eigen-motions of the combined walking and running model. One can see that the two activities are easily separated in this subspace. The walking components appear on the left of the plot and form a relatively dense set. By contrast, the running components are sparser because inter-subject variation is larger, indicating that more training examples are required for a satisfactory model.

While the motion models exhibit this inter-subject and inter-activity variation, we would not expect pure pose models to exhibit similar structure. For example to demonstrate this we also learned a pose model by applying PCA on individual poses in the same dataset. Fig. 5 shows poses from the walking data projected onto the first four eigen-directions of the subspace model learned from poses in the walking motions. It is clear that there is no inter-subject separation in the pose model.



(a) (b) (c)

Fig. 6. Sampling the single activity databases. In each plot we show the most probable sample that is at the origin, and a sample with very low probability (far from the origin) for the (a) walking, (b) running, and (c) golfing databases. Their respective motions are shown in Fig. 7. The dashed curves are one standard deviation ellipses for the underlying Gaussian density model for the data.

3.5 Model Convexity

PCA provides a subspace model within which motions are expressed as linear combinations of the eigen-motions (4). With probabilistic PCA [43] one further constrains the model with a multivariate Gaussian density. A key property of such linear models is the convexity of the motions, i.e., that linear combinations of motions (or eigen-motions) are legitimate motions.

While convexity is clearly violated with pose data (cf., Fig. 5a), we find that with the subspace motion models convexity is satisfied to a much greater extent. In other words, we find that random samples from the subspaces, according to a subspace Gaussian model for walking, running and the golf swing, all produce plausible motions. Fig. 7 depicts two motions from each of (a) the walking model, (b) the running model, and (c) the combined model. The first row in each case depicts the mean motion for each model, corresponding to the origin of the respective subspaces. As shown in Fig. 6 the origin is relatively far from any particular training motion, yet these motions look quite plausible. The second motion in each case corresponds to a point drawn at random that is far from the origin and any training motion (as shown in Fig. 6). These points, typical of other random samples from the underlying Gaussian density, also depict plausible motions. Accordingly, the models appear to generalize naturally to points relatively far from the training data.

The multi-activity model learned from the combined running and walking data does not exhibit the same property. Fig. 8 shows the subspace spanned by the first two eigen-motions of the combined model. In addition to the training data, the figure shows the locations of four points that lie roughly between the projections of the walking and running data. The four rows of Fig. 9 depict the corresponding motions (for which the remaining subspace coefficients, $\alpha_j = 0$, for $3 \leq j \leq m$). While three of the motions are plausible mixtures of running and walking, the top row of

Fig. 9 clearly shows an implausible motion. Here we find that points close to the training data generate plausible motions, but far from the training data the motions become less plausible.

Nevertheless there are regions of the subspace between walking and running data points that do correspond to plausible models. These regions facilitate transitions between walking and running that are essential if we wish to be able to track subjects through such transitions, as will be shown in section 6.

4 Tracking Framework

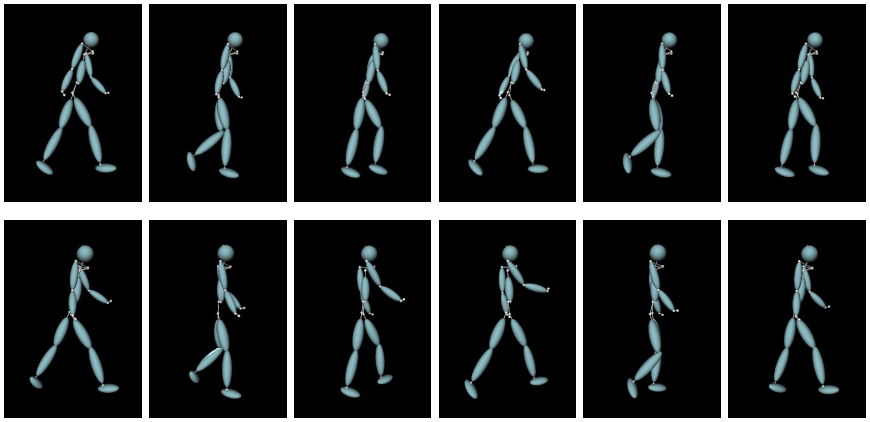
In this section we show how the motion models of Section 3 can be used for 3D people tracking. Our goal is to show that with activity-specific motion models one can often formulate and solve the tracking problem straightforwardly with deterministic optimisation. Here, tracking is expressed as a nonlinear least-squares optimization, and then solved using Levenberg-Marquardt [44].

The tracking is performed with a sliding temporal window. At each time instant t we find the optimal target parameters for f frames within a temporal window from time t to time $t+f-1$. Within this window, the relevant target parameters include the subspace coefficients, $\{\alpha_i\}_{i=1}^m$, the global position and orientation of the body at each frame $\{\mathbf{g}_j\}$ and the phases of the motion at each frame $\{\mu_j\}$, for $t \leq j < t+f$:

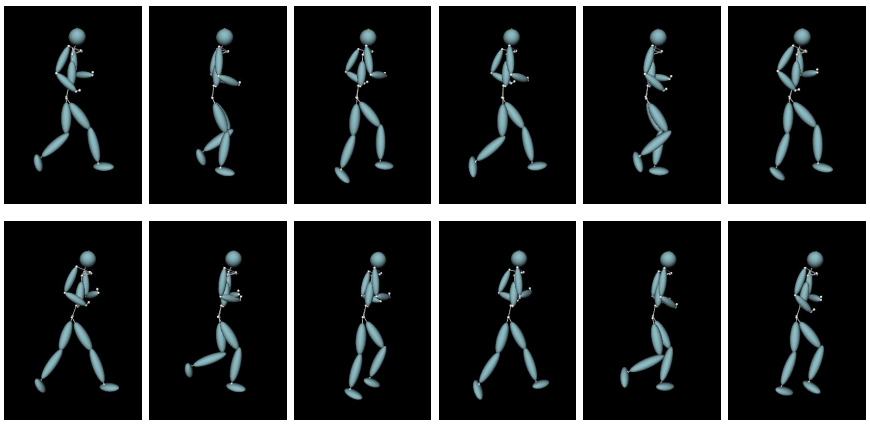
$$\mathbf{S}_t \;=\; [\alpha_1, \ldots, \alpha_m, \mu_t, \ldots, \mu_{t+f-1}, \mathbf{g}_t, \ldots, \mathbf{g}_{t+f-1}] \;\; . \tag{7}$$

While the global pose and phase of the motion vary throughout the temporal window, the unknown subspace coefficents are assumed to be constant over the window.

After minimizing an objective function over the unknown parameters $\mathbf{S}_t$, we extract the pose estimate at time t that is given by the estimated subspace coefficients $\{\hat{\alpha}_i\}$, along with the global parameters and phase at time t, i.e., $\hat{\mathbf{g}}_t$ and $\hat{\mu}_t$. Because the temporal estimation windows overlap from one time instant to the next, the estimated target parameters tend to vary smoothly over time. In particular, with such a sliding window the estimate of the pose at time t is effectively influenced by both past and future data. It is influenced by past data because we assume smoothness between parameters at time t and estimates already found at previous time instants $t-1$ and $t-2$. It is influenced by future data as data constraints on the motion are obtained from image frames at times $t+1$ through $t+f-1$.



(a)



(b)

(c)

Fig. 7. Sampling the first five components of each single activity database produce physically possible motions. The odd rows show the highest probability sample that for each single-motion database, which is the at the origin $\alpha_i = 0, \forall i$. The even rows show some low probability samples very far from the training motions to demonstrate that even those samples produce realistic motions. The coefficients for these motion are shown in Fig. 6 (a,b,c) respectively. **First two rows (a):** Walking, **Middle rows (b):** Running, **Last two rows (c):** Golf swing samples.

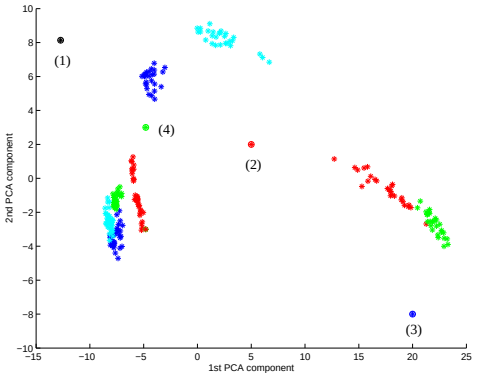


Fig. 8. Sampling the multi-activity subspace. The 4 samples that generate the motions of Fig. 9 are depicted.

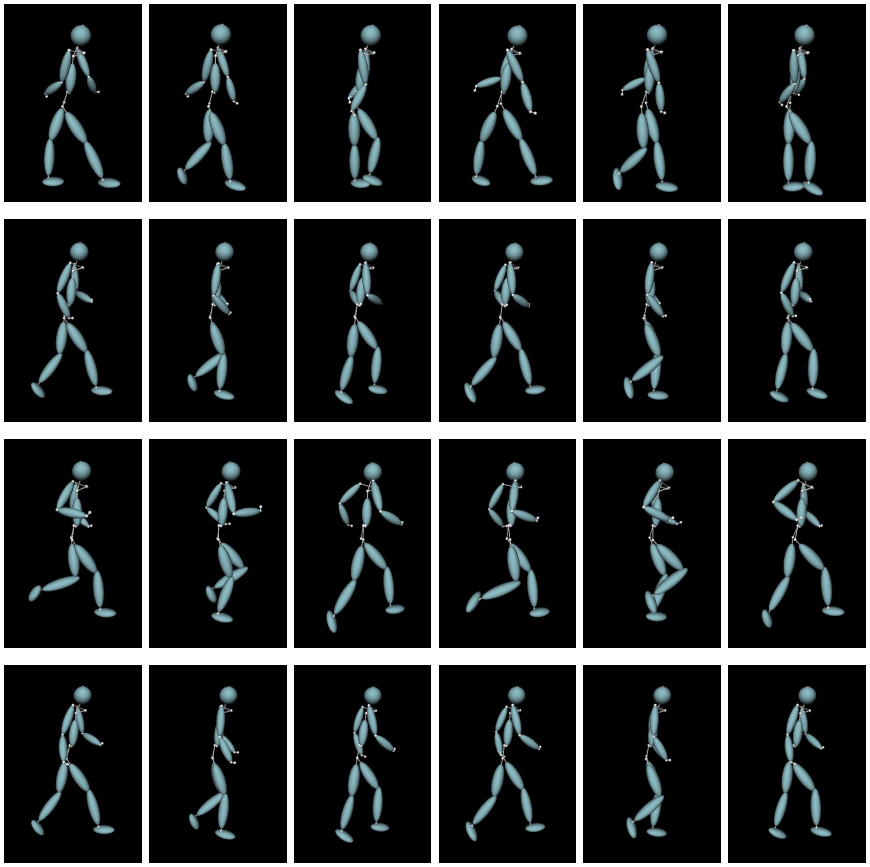


Fig. 9. Sampling the first two components of a multi-activity database compose of walking and running motions can produce physically impossible motions. The coefficients of the motions depicted in this figure are shown in Fig. 8. **Top row:** Physically impossible motion. The input motion space compose of walking and running is not convex. **Middle row:** Physically possible motion close to a walking. **Bottom row:** Motion close to a running. As the convexity of the input space is assumed when doing PCA, and it may not be the case, the resulting motion as a combination of principal directions can be physically impossible.

4.1 Objective Function

We use the image data to constrain the target parameters with a collection of n_{obs} constraint equations of the form

$$O(\mathbf{x}_i, \mathbf{S}) \;=\; \epsilon_i \;\;, \quad 1 \leq i \leq n_{\text{obs}} \;\;, \tag{8}$$

where the $\mathbf{x}_i$ are 2D image features, O is a differentiable function whose value is zero for the correct value of $\mathbf{S}$ and noise-free data, and ϵ_i denotes the residual error in the i^{th} constraint. Our objective is to minimize the sum of the squared constraint errors. Because some measurements may be noisier than others, and our observations may come from different image properties that might not be commensurate with one another, we weight each constraint of type $type$ with a constant, w^{type}. In effect, this is equivalent to a model in which the constraint residuals are IID Gaussian with isotropic covariance, and the weights are just inverse variances. In practice, the values of the different w^{type} are chosen heuristically based on the expected errors for each type of observation.

Finally, since image data are often noisy, and sometimes underconstrain the target parameters, we further assume regularization terms that encourage smoothness in the global model. We also assume that the phase of the motion varies smoothly. The resulting total energy to be minimized at time t, F_t, can therefore be expressed as

$$F_t = F_{o,t} + F_{g,t} + F_{\mu,t} + F_{\alpha,t} \tag{9}$$

with

$$\begin{aligned} F_{o,t} &= \sum_{i=1}^{nobs} w^{type_i} \left\| O^{type_i}(\mathbf{x}_i, \mathbf{S}) \right\|^2 \;\;, \quad F_{g,t} = w_G \sum_{j=t}^{t+f-1} \left\| \mathbf{g}_j - 2\mathbf{g}_{j-1} + \mathbf{g}_{j-2} \right\|^2 , \\ F_{\mu,t} &= w_\mu \sum_{j=t}^{t+f-1} \left(\mu_j - 2\mu_{j-1} + \mu_{j-2} \right)^2 \;\;, \quad F_{\alpha,t} = w_\alpha \sum_{i=1}^{m} (\alpha_i - \hat{\alpha}_i)^2 \;\;, \end{aligned} \tag{10}$$

where O^{type} is the function that corresponds to a particular observation type, w_G, w_μ and w_α are scalar weights, and $\hat{\alpha}_i$ denote the subspace coefficients estimated in the previous window of f frames at time $t-1$. The value of f is chosen to be sufficiently large to produce smooth results; in practice we use $f = 3$. Finally, in (10), the variables $\mathbf{g}_{t-1}$, $\mathbf{g}_{t-2}$, μ_{t-1} and μ_{t-2} are taken to be the values estimated from previous two time instants, and are therefore fixed during estimation at time t.

Minimizing F_t using the Levenberg-Marquardt algorithm [44] involves computing its Jacobian with respect to the elements of the state vector $\mathbf{S}$. Since the O functions of Eq. 10 are differentiable with respect to the elements of $\mathbf{S}$, computing the

derivatives with respect to the $\mathbf{g}_t$ is straightforward. Those with respect to the α_i and μ_t can be written as

$$\frac{\partial F_t}{\partial \alpha_i} = \sum_{k=t}^{t+f-1} \sum_{j=1}^{D} \frac{\partial F_{o,t}}{\partial \theta_j^k} \cdot \frac{\partial \theta_j^k}{\partial \alpha_i} + \frac{\partial F_{\alpha,t}}{\partial \alpha_i} \quad , \tag{11}$$

$$\frac{\partial F_t}{\partial \mu_k} = \sum_{j=1}^{D} \frac{\partial F_{o,t}}{\partial \theta_j^k} \cdot \frac{\partial \theta_j^k}{\partial \mu_k} + \frac{\partial F_{\mu,t}}{\partial \mu_k} \quad , \tag{12}$$

where the θ_j^k represents the vector of individual joint angles at phase μ_k, defined as the j'th component of $\psi(\mu_k, \alpha_1, \cdots, \alpha_m)$ in Eq. 6. The derivatives of F_t with respect to the D individual joints angles $\frac{\partial F_{o,t}}{\partial \theta_j^k}$ can be easily computed [45]. Because the θ_j^k are linear combinations of the Θ_{ij}^k eigen-poses, $\frac{\partial \theta_j^k}{\partial \alpha_i}$ reduces to Θ_{ij}^k, the jth coordinate of Θ_i^k. Similarly, we can write

$$\frac{\partial \theta_j^k}{\partial \mu_k} = \sum_{i=1}^{m} \alpha_i \frac{\partial \Theta_{ij}^k}{\partial \mu_k} \quad , \tag{13}$$

where the $\frac{\partial \Theta_{ij}^k}{\partial \mu_t}$ can be evaluated using finite differences and stored when building the motion models, as depicted in Fig. 2.

Recall that for cyclic motions such as walking and running, the phase is periodic and hence the second order prediction $\mu_{j-1} - \mu_{j-2}$ should be taken $\mod 1$ in Eq. 9. This allows the cyclic tracking sequences to be arbitrarily long, not just a single cycle. Of course, one can also track sequences that comprise fractional parts of either cyclic or acyclic motion model.

The weights w in Eq. (10) were set manually, but their exact values were not found to be particularly critical. In some experiments the measurements provided sufficient strong constraints that the smoothness energy terms in Eq. (10) played a very minor role; in such cases the values of w_G, w_μ and w_α could be set much smaller than the weights on the measurement errors in $F_{o,t}$. Nonetheless, for each set of experiments below (i.e, those using the same types of measurements), the weights were fixed across all input sequences.

4.2 Computational Requirements

The fact that one can track effectively with straight-forward optimization means that our prior motion models greatly constrain the inference problem. That is, the resulting posterior distributions are not so complex (e.g., multimodal) that one must use computationally demanding inference methos such as sequential Monta Carlo or particle filtering.

Monte Carlo approaches, like that in [8], rely on randomly generating particles and evaluating their fitness. Because the cost of creating particles is negligible, the main cost of each iteration comes from evaluating a log likelihood, such as F_t in (9), for each particle. In a typical particle filter, like the Condensation algorithm [7], the number of particles needed to effectively approximate the posterior on a D-dimensional state space grows exponentially with D [5,46]. With dimensionality reduction, like that obtained with the subspace motion model, the state dimension is greatly reduced. Nevertheless, the number of particles required can still be prohibitive [8].

By contrast, the main cost at each iteration of our deterministic optimization scheme comes from evaluating F_t and its Jacobian. In our implementation, this cost is roughly proportional to $1 + \log(D)$ times the cost of computing F_t alone, where D is the number of joint angles of (12). The reason this factor grows slowly with D is that the partial derivatives, $\frac{\partial F_t}{\partial \theta_j}$, which require most of the computation, are computed analytically and involve many intermediate results than can be cached and reused. As a result, with R iterations per frame, the total time required by our algorithm is roughly proportional $R(1 + \log(D))$ times the cost of evaluating F_t. Since we use a small number of iterations, less than 15 for all experiments in this paper, the total cost of our approach remains much smaller than typical probabilistic methods. The different experiments run in this paper took less than a minute per frame, with a non-optimized implementation.

5 Monocular Tracking

We first demonstrate our approach in the context of monocular tracking [47]. Since we wish to operate outdoors in an uncontrolled environment, tracking people wearing normal clothes, it is difficult to rely solely on any one image cue. Here we therefore take advantage of several sources of information.

5.1 Projection Constraints

To constrain the location of several key joints, we track their approximate image projections across the sequence. These 2D joint locations were estimated with a 2D image-based tracker. Figure 10 shows the 2D tracking locations for two test sequences; we track 9 points for walking sequences, and 6 for the golf swing.

For joint j, we therefore obtain approximate 2–D positions $\mathbf{x}^{\mathbf{j}}$ in each frame. From the target parameters $\mathbf{S}$ we know the 3–D position of the corresponding joint. We then take the corresponding constraint function, $O^{joint}(\mathbf{x}^{\mathbf{j}}, \mathbf{S})$, to be the 2–D Euclidean distance between the joint's projection into the image plane and the mea-

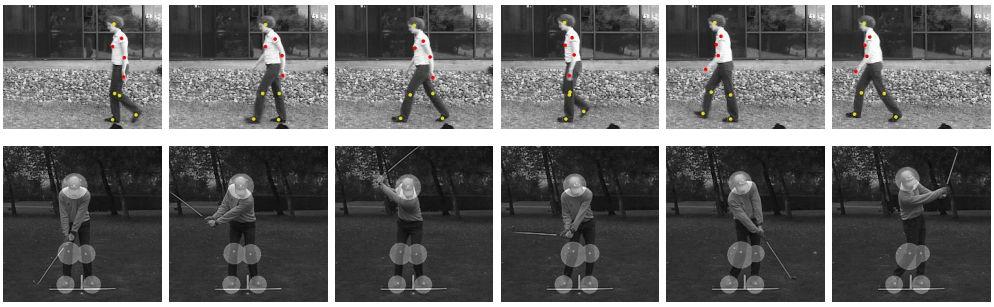


Fig. 10. 2D Tracking using the WSL tracker. **Top row:** Tracking the chest, knees, head, ankles and visible arm. The tracked upper body joints are shown in red, with the head and tracked lower joints points shown in yellow. **Bottom row**: Regions used for tracking the ankles, knees, and head are shown.

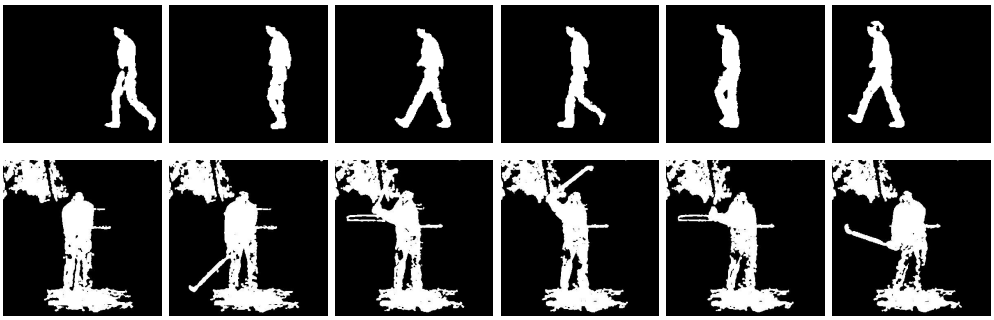


Fig. 11. Poor quality foreground binary mask. **First row:** Extracted from the walking sequence of Fig. 12 and **Second row:** from the golf swing of Fig. 17.

surement of its 2-D image position.

5.2 Foreground and Background

Given an image of the background without the subject, we can extract rough binary masks (silhouettes) of the foreground, like those in Fig. 11. Because the background in our video is not truly static the masks are expected to be noisy. Nevertheless, they can be exploited as follows. We randomly sample the binary mask, and for each sample $\mathbf{x_i}$ we define a *Background/Foreground function* $O^{fg/bg}(\mathbf{x_i}, \mathbf{S})$ that is 0 if the line of sight through $\mathbf{x_i}$ intersects the model. Otherwise, it is equal to the 3D distance between the line of sight and the nearest model point. In other words, $O^{fg/bg}$ is a differentiable function that introduces a penalty for each point in the foreground binary mask that does *not* back-project onto the model.

Minimizing $O^{fg/bg}$ in the least squares sense tends to maximize the overlap between the model's projection and the foreground binary mask. This helps to prevent target drift.

5.3 Point Correspondences (Optical Flow)

We use 2–D point correspondences in pairs of consecutive images as an additional source of information: We project the 3–D model into the first image of the pair. We then sample image points to which the model projects and use a normalized cross-correlation algorithm to compute displacements of these points from that frame to the next. This provides us with measurement pairs of corresponding points in two consecutive frames, $\mathbf{p}_i = (\mathbf{x}_i^1, \mathbf{x}_i^2)$. The correspondence penalty function, $O^{corr}(\mathbf{p}_i, \mathbf{S})$ is given as follows: We back-project $\mathbf{x}_i^1$ to the 3–D model surface and reproject it to the second image. We then take $O^{corr}(\mathbf{p}_i, \mathbf{S})$ to be the Euclidean distance in the image plane between this reprojection and corresponding $\mathbf{x}_i^2$.

5.4 Experimental Results

We test our tracker on real and synthetic data. In each case the use of prior motion models is crucial; without the motion models the tracker diverge within a few frames in every experiment.

5.4.1 Real data

The results shown here 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. For each test sequence we manually initialize the position and orientation of the root node of the body in the first frame so that it projects approximately to the right place.

We also manually specify the 2D locations of the joints to be tracked by WSL [48]. 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 structure. By identifying stable properties of appearance the tracker can weight them more heavily for motion estimation, while less stable properties can be proportionately down-weighted. This gives it robustness to partial occlusions. In the first frame we specified 9 points that we wish to track, namely, the ankles, knees, chest, head, left shoulder, elbow and hand.

This entire process requires only a few mouse clicks and could easily be improved by using automated posture detection techniques (e.g., [20,26,21,22,24]). Simple methods were used to detect the key postures defined in Section 3 for each sequence. Using spline interpolation, we assign an initial value for μ_t for all the frames in the sequence, as depicted in Figs. 13b and 16b. Finally, the motion is ini-

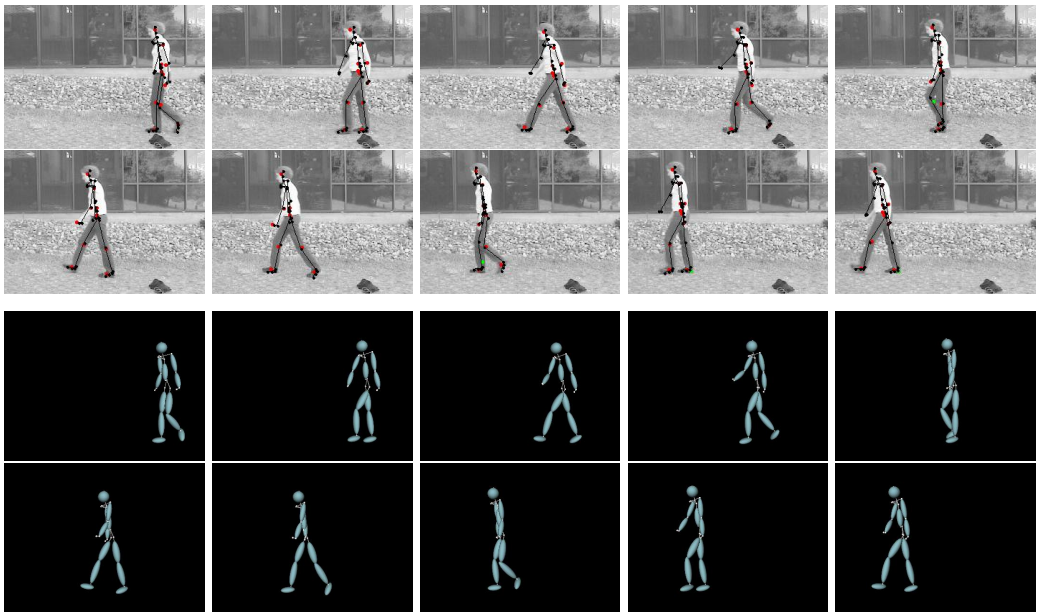


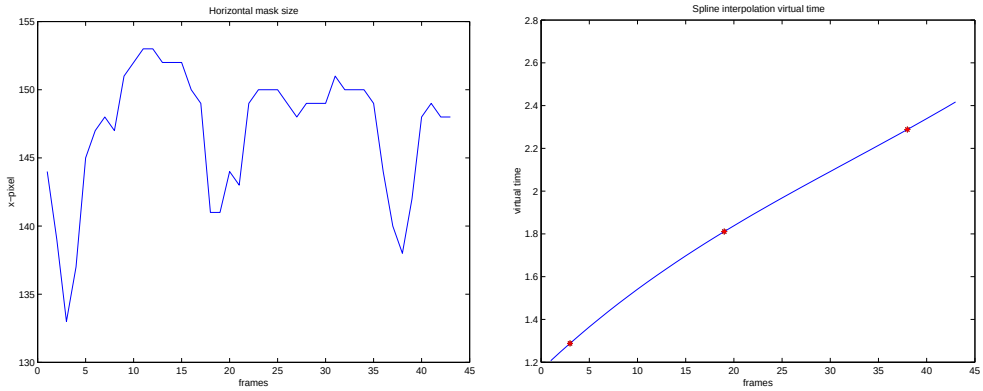
Fig. 12. Monocular Tracking of a 43 frames walking motion. **First two rows**: The skeleton of the recovered 3D model is projected onto the images. **Bottom two rows**: Volumetric primitives of the recovered 3D model projected into a similar view.

tially taken to be the mean motion Θ_0, i.e., the subspace coefficients α_i are initially set to zero. Given these initial conditions we minimize F_t in (9) using Levenberg-Marquardt.

5.4.1.1 Walking Fig. 12 shows a well-known walking sequence [8,49,50]. To perform the 2D tracking we used a version of the WSL tracker [48]. To initialize the phase parameter, μ_t, we used a simple background subtraction method to compute foreground masks (e.g., see Fig. 11). Times at which the mask width was minimal were taken to be the times at which the legs were together (i.e., $\mu_t = 0.25$ or $\mu_t = 0.75$). Spline interpolation was then used to approximate μ_t at all other frames in the sequence (see Fig. 13b). More sophisticated detectors [20–22,24] would be necessary in more challenging situations, but were not needed here.

The optimal motion found is shown in Figure 12. There we show the estimated 3D model projected onto several frames of the sequence. We also show the rendered 3D volumetric model alone. The tracker was successful, producing a 3D motion that is plausible and well synchronized with the video. The right (occluded) arm was not tracked by the WSL tracker, and hence was only weakly constrained by the objective function. Note that even though it is not well reconstructed by the model (does not fit the image data), it has a plausible rotation.

5.4.1.2 Golf Swing As discussed in Section 3.3, the golf swings used to train the model were *full swings* from the CMU database. They were performed by nei-



(a) (b)

Fig. 13. Automatic Initialization of the virtual time parameter μ_t for the walking sequence of Fig. 12. (a) Width of the detected silhouette. (b) Spline interpolation for the detected key-postures.

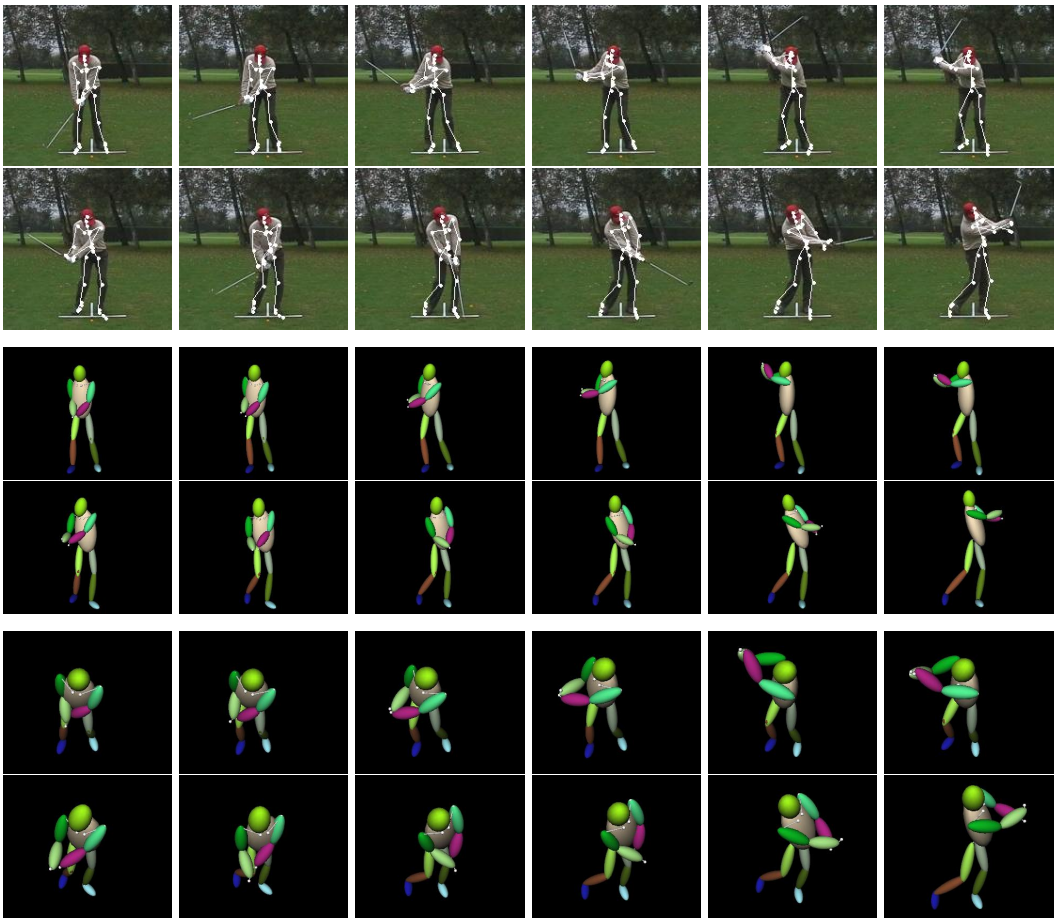
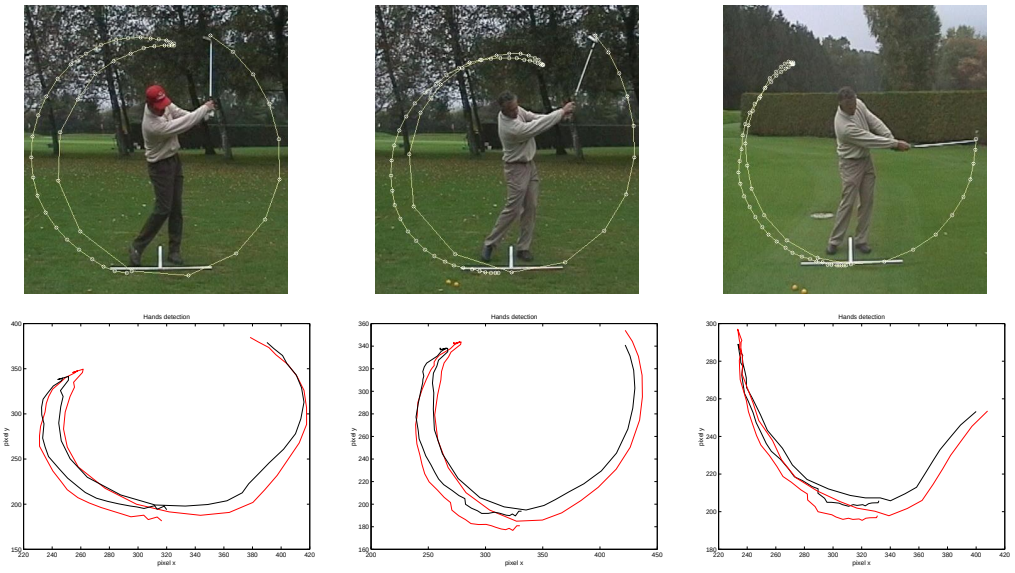


Fig. 14. Monocular Tracking a full swing in a 45 frame sequence. **First two rows**: The skeleton of the recovered 3–D model is projected into a representative subset of images. **Middle two rows**: Volumetric primitives of the recovered 3–D model projected into the same views. **Bottom two rows**: Volumetric primitives of the 3–D model as seen from above.



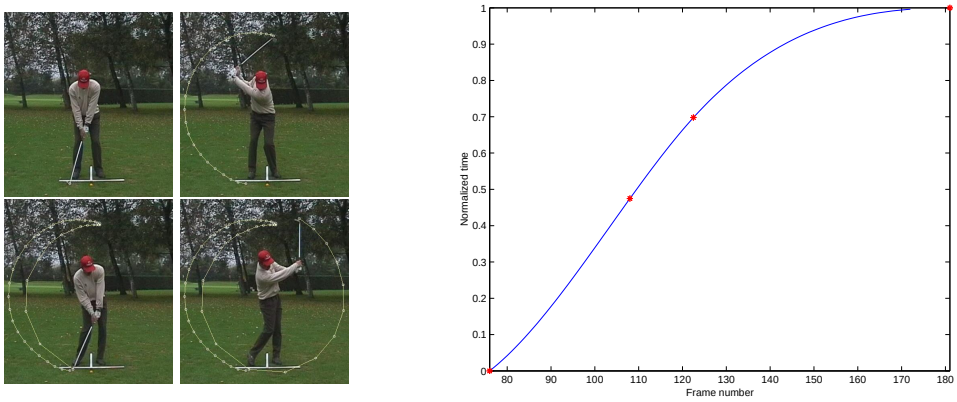
(a) (b) (c)

Fig. 15. Detected hand trajectories for the *full* swing in Fig. 14 and the *approach* swing in Fig. 18. The left and right hand positions (pixel units) are represented in black and red respectively.

ther of the golfers shown in Figs. 14, 17 and 18. With the WSL tracker we tracked five points on the body, namely, the knees, ankles and head (see Fig. 10). Because the hand tends to rotate during the motion, to track the wrists we have found it more effective to use a club tracking algorithm [51] that takes advantage of the information provided by the whole shaft. Its output is depicted by the first row of Fig. 15, and the corresponding recovered hand trajectories by the second row. 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. Hypotheses on the position are first generated by detecting pairs of close parallel line segments in the frames, and then robustly fitting a 2D motion model over several frames simultaneously. From the recovered club motion, we can infer the 2D hand trajectories of the bottom row of Fig. 15.

For each sequence, we first run the golf club tracker [51]. As shown in Fig. 16a, for each sequence, the detected club positions let us initialize the phase parameters by telling us in which four frames the key postures of Fig. 4 can be observed. With the times of the key postures, spline interpolation is then used to assign a phase to all other frames in the sequence (see Fig. 16b). As not everybody performs the motion at the same speed, these phases are only initial guesses, which are refined during the actual optimization. Thus the temporal alignment does not need to be precise, but it gives a rough initialization for each frame.

Figures 14 and 17 show the projections of the recovered 3D skeleton in a representative subset of images of two *full* swings performed by subjects whose motion



(a) (b)

Fig. 16. Assigning normalized times to the frames of Fig. 14. (a) We use the automatically detected club positions to identify the key postures of Fig. 4. (b) The corresponding normalized times are denoted by red dots. Spline interpolation is then used to initialize the μ_t parameter for all other frames in the sequence.

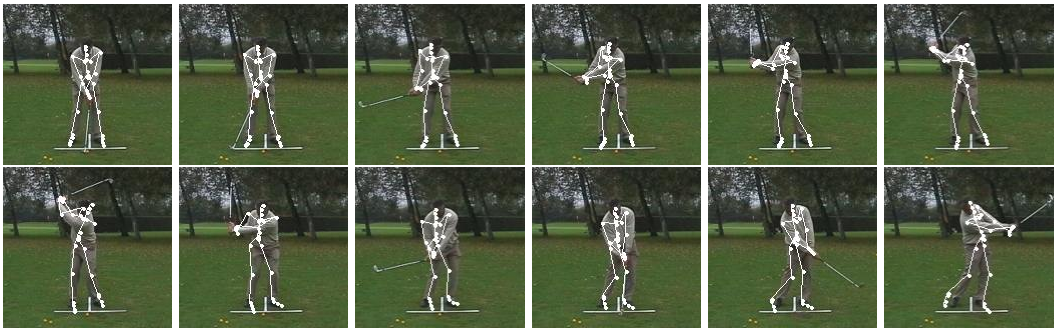


Fig. 17. Monocular tracking a 68 frame swing sequence. The skeleton of the recovered 3–D model is projected onto the images.

was not used in the motion database. Note the accuracy of the results. Figure 18 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. 14 and from the swing used to train the system. The club does not go as high and, as shown in Fig. 15, the hands travel a much shorter distance. As shown by the projection of the 3D skeleton, the system has enough flexibility to generalize to this motion. Note, however, that the right leg bends too much at the end of the swing, which is caused by the small number of training motions and the fact that every training swing exhibited the same anomoly. A natural way to avoid this problem in the future would be to use a larger training set with a greater variety of motions.

Finally, Fig. 19 helps to show that the model has sufficient flexibility to do the *wrong* thing given insufficient image data. That is, even though we use an activity-specific motion model, the problem is not so constrained that we are guaranteed to get valid postures or motions without using the image information correctly.

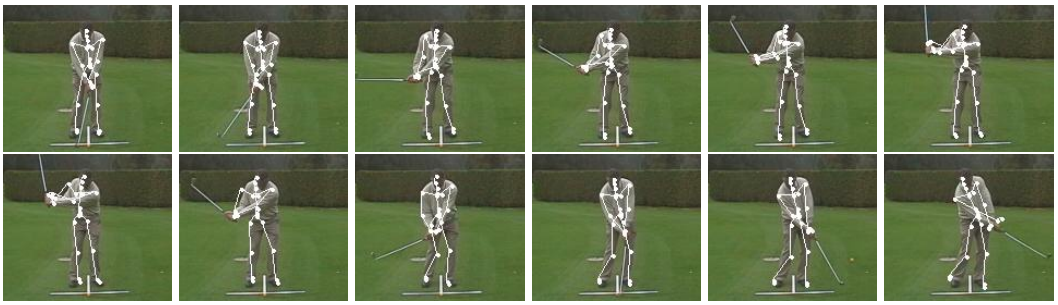


Fig. 18. Monocular Tracking an approach swing during which the club goes much less high than in a driving swing. The skeleton of the recovered 3–D model is projected onto the images.

(a) (b) (c) (d) (e) (f)

Fig. 19. Tracking using only joint constraints vs using the complete objective function. (a) Original image. (b) 2–D appearance based tracking result. (c) 2–D projection of the tracking results using only joint constraints. The problem is under-constrained and a multiple set of solutions are possible. (d) 3–D tracking results using only joint constraints. (e) and (f) The set of solutions is reduced using correspondences.

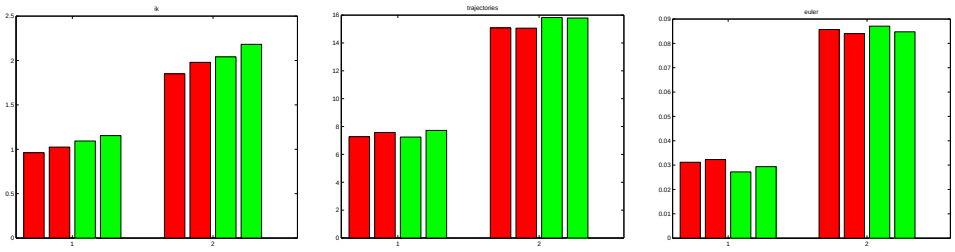
5.4.2 Synthetic data

We projected 3D motion capture data using a virtual camera to produce 2D joint positions that we then use an input to our tracker. The virtual camera is such that the projections fall within a 640x480 virtual image, with the root projecting at the center of the image. We initialized the phase of the motion μ_t to a linear function, 0 at the beginning and 1 at the end of the sequence. The style of the motion was initialized to be the mean motion. Both μ_t and the $\{\alpha_i\}$ were refined during the tracking.

We used temporal windows of sizes 3 and 5, with very similar results, as shown in Fig. 20. We also tested the influence of the number of 2D joint positions given as input to the tracker, by using the whole set of joints, or the same subset of joints used to track the sequence of Fig. 12, namely, the ankles, knees, chest, head, shoulder, elbow and hand. Both cases result in very similar accuracy, as depicted in Fig. 20. The errors, as expected, are bigger when tracking testing data than training data. Note that the tracker is very accurate, the 3D errors are 0.7 cm per joint in mean for the training sequences and 1.5 cm in mean for the testing sequences.

It is also of interest to test the sensitivity of the tracker to the relative magnitudes of the smoothness and observation weights in Eq. (10). Fig. 21 shows the results of tracking synthetic training and testing sequences with different values of w_{type}/w_s,

	Training			Test		
	2D proj.	3D loc.	Euler	2D proj.	3D loc.	Euler
All, $f = 3$	0.960	7.271	0.031	1.849	15.079	0.086
Subset, $f = 3$	1.024	7.575	0.0322	1.979	15.062	0.0839
All, $f = 5$	1.093	7.246	0.0272	2.041	15.823	0.087
Subset, $f = 5$	1.153	7.721	0.0293	2.182	15.791	0.0847



2D projection 3D location Euler angles

Fig. 20. **Tracking mean errors as a function of the window size and the number of 2D constraints.** Three types of errors (2D projection (pixels), 3D location (mm), Euler angles (radians) are depicted. Each plot is split in two groups, the left one represents errors when tracking training data and the right one test data. For each group 4 error bars of 2 different colors are depicted, each color represents a different window size (3 on red, and 5 on green). For each color two bars show the errors first for the complete set of joints and then for the subset of joints, with similar results. Note that the estimated 3D joint location errors are very small, 0.7 cm in mean for the training data sequences, and 1.5 cm for the testing ones.

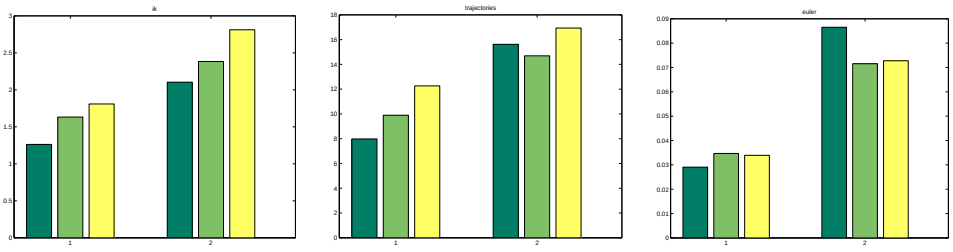
ranging from 0.1 to 10, with $w_s = w_g = w_\mu = w_\alpha$. All experiments yielded similar results, indicating that the tracker is not particularly sensitive to these parameters.

5.4.3 Failure Modes

We have demonstrated that the tracking works well for different cyclic (walking) and a-cyclic motions (golfing). The tracked motions are different from the ones used for training, but remain relatively close. In this section we use a caricatured walking sequence to test when the generalization capabilities of our motion models fail. The caricatured walking is very different from the motions used for training, and the PCA-based motion models do not generalize to this motion well. The style coefficients recovered by the tracker are very far from the training ones (at least 6 standard deviations), resulting in impossible poses as depicted by Fig. 22, when using a 3 or 5 frame temporal window.

When using PCA-based motion models, one should track motions that remain relatively close to the training data, since the only motions that the tracker is capable of producing are those in the subspace. In case other motions are wanted to be

	Training			Test		
	2D proj.	3D loc.	Euler	2D proj.	3D loc.	Euler
$w_{type}/w_s = 0.1$	1.262	7.979	0.0291	2.104	15.621	0.0865
$w_{type}/w_s = 1$	1.632	9.895	0.0347	2.384	14.692	0.0715
$w_{type}/w_s = 10$	1.808	12.263	0.0339	2.812	16.934	0.0728



2D projection 3D location Euler angles

Fig. 21. **Tracking mean errors as a function of the weights.** Tracking results are given for experiments with three different types of measurement errors (2D projection (pixels), 3D location (mm), Euler angles (radians)). Each plot is split in two groups, the left one represents errors when tracking training data and the right one for tracking test data. For each group 3 error bars of different colors are depicted. Each color represents different relative weights (dark green $w_{type}/w_s = 0.1$, green $w_{type}/w_s = 1$, and yellow $w_{type}/w_s = 10$), with $w_s = w_g = w_\mu = w_\alpha$. Note that tracker is not very sensitive to the specific value of the weights.

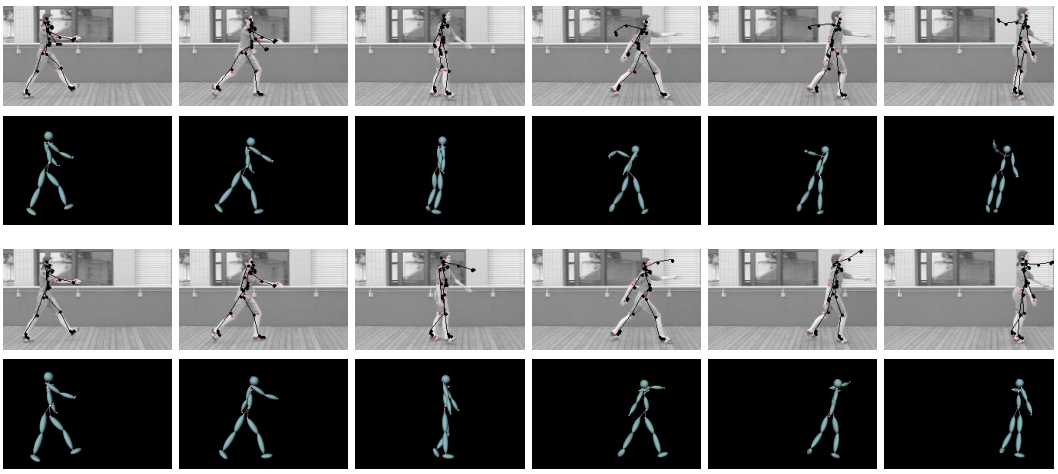


Fig. 22. Tracking 40 frames of an exaggerated gait. **First two rows:** 3 frame window. **Last two rows:** 5 frame window. The tracker results in impossible positions.

tracked, one should include examples of such motions when learning the models, or apply other techniques such as Gaussian Processes (GP) [34] that have better generalization properties.

6 Multi-view Tracking

When several synchronized video streams are available, we use a correlation-based stereo algorithm [52] to extract a cloud of 3–D points at each frame, to which we fit the motion model.

6.1 Objective Function

Recall from Section 3 that we represent the human body as a set of volumetric primitives attached to an articulated 3–D skeleton. For multi-view tracking we treat them as implicit surfaces as this provides a differentiable objective function which can be fit to the 3D stereo data while ignoring measurement outliers. Following [45] the body is divided into several body parts; each body part b includes n_b ellipsoidal primitives attached to the skeleton. Associated with each primitive is a field function f_i, of the form

$$f_i(\mathbf{x}, \mathbf{S}) = b_i \exp(-a_i d_i(\mathbf{x}, \mathbf{S})) \quad , \tag{14}$$

where $\mathbf{x}$ is a 3–D point, a_i, b_i are constant values, d_i is the algebraic distance to the center of the primitive, and $\mathbf{S}$, is the state vector in (7). The complete field function for body part b is taken to be

$$f^b(\mathbf{x}, \mathbf{S}) = \sum_{i=1}^{n_b} f_i(\mathbf{x}, \mathbf{S}) \quad , \tag{15}$$

and the skin is defined by the level set

$$\mathcal{S}K(\mathbf{x}, \mathbf{S}) = \bigcup_{b=1}^{B} \{\mathbf{x} \in \mathcal{R}^3 | f^b(\mathbf{x}, \mathbf{S}) = C\} \tag{16}$$

where C is a constant, and B is the total number of body parts. A 3D point $\mathbf{x}$ is said attached to body part b if

$$b = \arg \min_{1 \le i \le B} |f^i(\mathbf{x}, \mathbf{S}) - C| \tag{17}$$

For each 3D stereo point, $\mathbf{x}_i$, we write

$$O^{stereo}(\mathbf{x}_i, \mathbf{S}) = f^b(\mathbf{x}_i, \mathbf{S}) - C \ . \tag{18}$$

Fitting the model to stereo-data then amounts to minimizing (9), the first term of which becomes

$$\sum_{j=t}^{t+f-1} \sum_{b=1}^{B} \sum_{\mathbf{x}_i \in b} (f^b(\mathbf{x}_{i,j}, \mathbf{S}) - C)^2 \ , \tag{19}$$

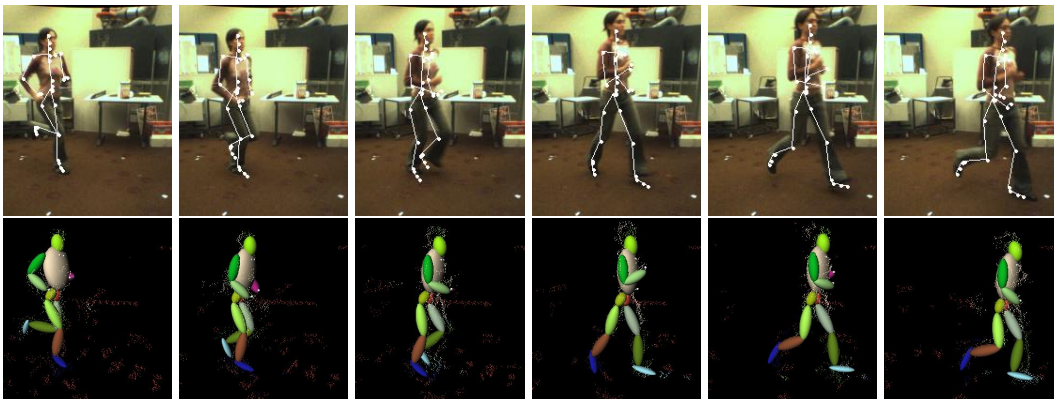


Fig. 23. Tracking a running motion. The legs are now correctly positioned in the whole sequence.

where $\mathbf{x}_{i,j}$ is a 3D stereo point belonging to frame j. Note that O^{stereo} is differentiable and its derivatives can be computed efficiently [45].

6.2 Experimental Results

We use stereo data acquired using a Digiclops^{tm} operating at a 640×480 resolution and a 14Hz frame rate. Because the frame rate is slow, the running subject of Fig. 23 remains within the capture volume for only 6 frames. The data shown in Fig. 24 are noisy and have low resolution for two reasons. First, to avoid motion blur, we used a high shutter speed that reduced exposure. Second, because the camera was fixed and the subject had to remain within the capture volume, she projected onto a small region of the image during the sequences. Of course, the quality of this stereo data could have been improved by using more sophisticated equipment. Nevertheless, our results show that the tracker is robust enough to exploit data acquired with cheap sensors.

Initially, the motion subspace coefficients are set to zero, as above. We manually initialized the phase of the motion μ_t in the first and last frame of the sequence. These points were then interpolated to produce an initial phase estimate in every frame. The initial guess does not have to be precise because the tracking does not work directly with the images but with the 3D data.

Fig. 25 shows results on walking sequences performed by two subjects whose motion capture data were also used as training data for the motion models. One can see from the figures that the legs are correctly positioned. The errors in the upper-body are caused by the large amount of noise in the stereo data.

Figure 26 depicts results from a walking sequence with a subject whose motion was not included in the training data. In this case he was also wearing four gyroscopes on his legs, one for each sagittal rotation of the hip and knee joints. The angular

Fig. 24. Input stereo data for the running sequence of Fig. 23. Side views of the 3–D points computed by the Digiclops ^{tm} system. Note that they are very noisy and lack depth because of the low quality of the video sequence.

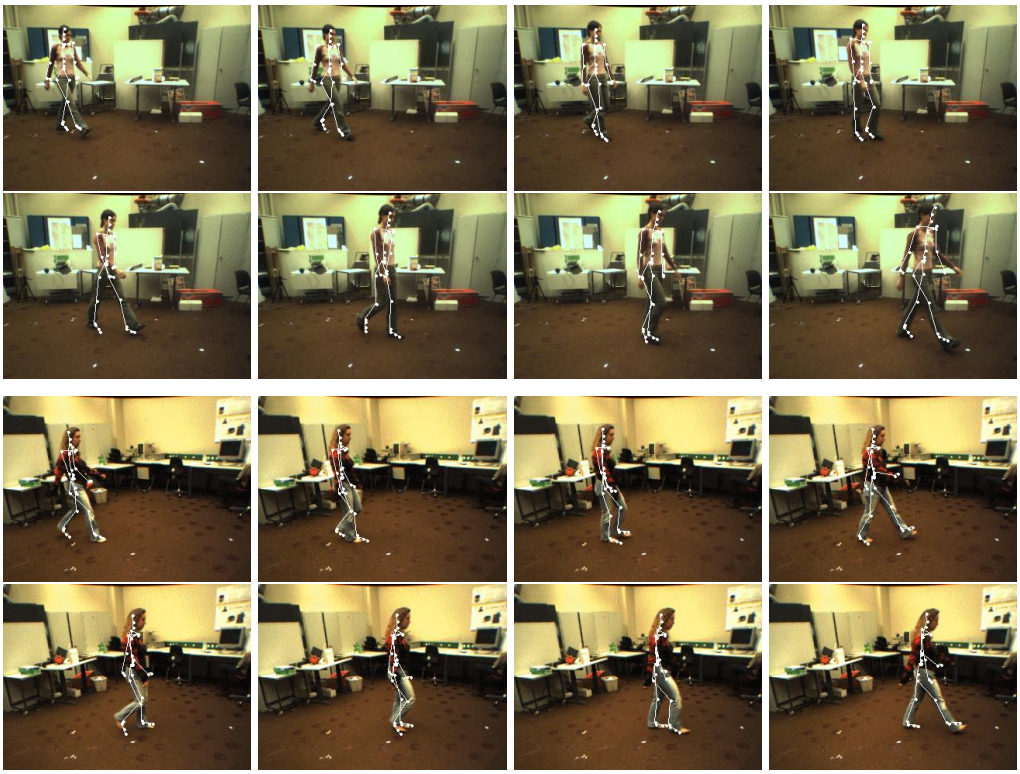


Fig. 25. Using low resolution stereo data to track the two women whose motions were not used to learn the motion model. The recovered skeleton poses are overlaid in white.

speeds they measured were used solely for comparison purposes. Their output was integrated to yield the absolute angles shown as dotted curve in Fig. 27. We overlay on these plots the values recovered by our tracker, showing that they are close, even though the left leg is severely occluded. Given the position of the visible leg, the PCA motion model constrains the occluded one to be in a plausible position close to the real one.

Figure 23 shows results for the running sequence of Fig. 24 using the running motion model. The pose of the legs is correctly recovered. The upper body tracking remains relatively imprecise because average errors in the stereo data are larger

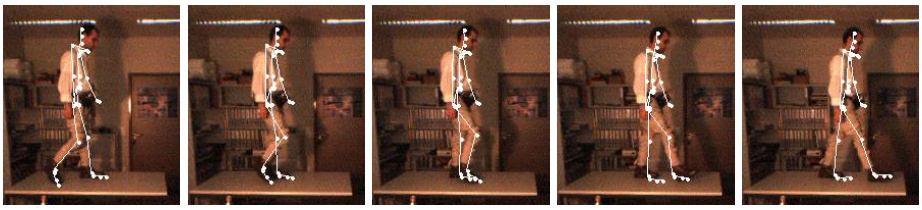


Fig. 26. Tracking a walking motion from a subject whose motion was not recorded in the database. The legs are correctly positioned.

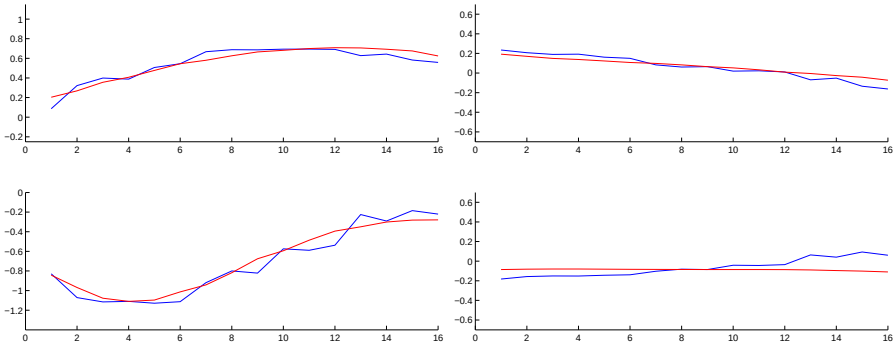


Fig. 27. Comparing recovered rotation angles using visual tracking (solid curve), and by integrating gyroscopic data (smooth curve) for the walk of Fig. 26. **Left column**: Right hip and knee sagittal rotations. **Right Column**: Same thing for the left leg. Note that both curves are very close in all plots, even though the left leg is severely occluded.

than the distance between the torso and the arms. Improving this would require the use of additional information, such as silhouettes. Here we restrict ourselves to stereo data to show that our framework can be used with very different objective functions.

Having a set of subspace coefficients per frame gives the system the freedom to automatically evolve from one activity to another. To demonstrate this we used our motion model learned for the combined running and walking data to track a transition from walking to running (see Fig. 28). In the first few frames the subject is walking, then for a couple of frames she performs the transition and runs for the rest of the sequence. The arms are not tracked because we focus on estimating the motion parameters of the lower body only. Here again, the legs are successfully tracked with small errors in foot positioning that are due to the fact that ankle flexion is not part of the motion database.

6.3 Recognition

The motion style is encoded by the subspace coefficients in (4). They measure the deviation from the average motion along orthogonal directions. Recall that during tracking, the subspace coefficients are permitted to vary from frame to frame. For recognition, we further reconstruct the 3D motion of the person with a single set

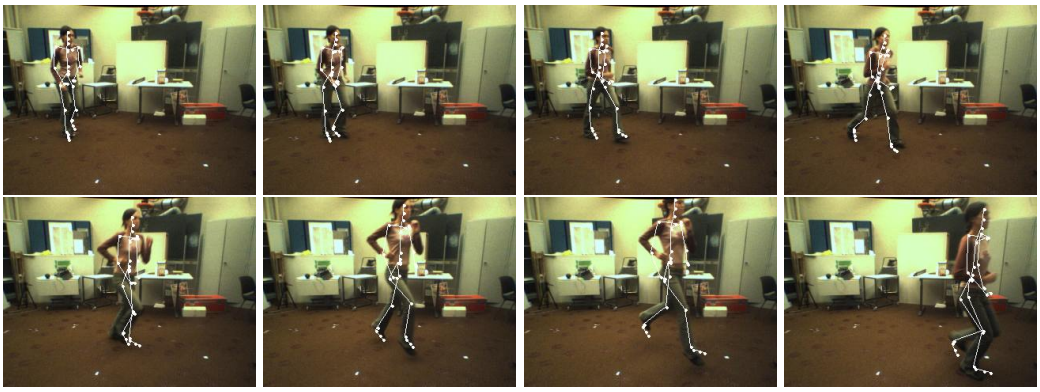


Fig. 28. Tracking the transition between walking and running. In the first four frames the subject is running. The transition occurs in the following three frames and the sequence ends with running. The whole sequence is shown.

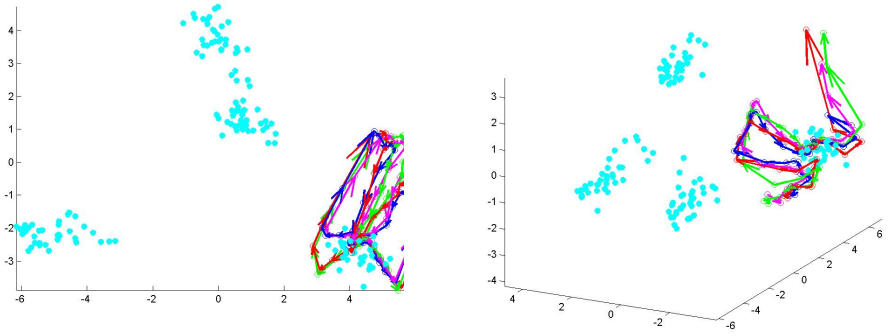
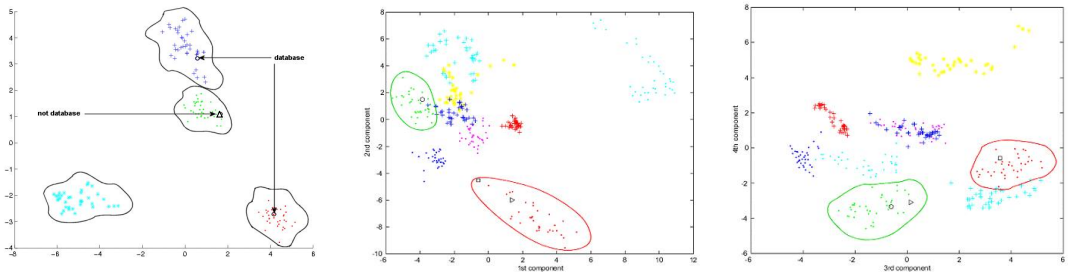


Fig. 29. Style coefficents, α_i, obtained when tracking a training sequence. The training data is shown in cyan. Different colors show different window sizes and number of 2D joint constraints.

of subspace coefficients for the entire sequence [53]. The reason is that we want to recover an average motion style during the sequence. Moreover, the estimate of the style coefficients is more reliable if we increase the number of poses we use to obtain it If we allow the style parameters to vary from frame to frame the style estimation is noisier, but the tracker is typically more accurate. This is illustrated in Fig. 29, when tracking with ground truth data and varying the subspace coefficients. Note that although the coefficients are close to the ones of that subject, their variance is relatively large.

The tracking algorithm used for recognition is divided into two steps. First, the normalized time μ_t and the global motion $\mathbf{g}_t$ are optimized frame by frame, assuming a constant style equal to the mean motion Θ_0. This provides a good initial estimate for a second step, where a global optimization is performed. In the global fit, the normalized time and global motion parameters are allowed to vary in every frame, but only one set subspace coefficients is used to represent the entire motion sequence. This is equivalent to minimizing (9), where the size of the sliding window



(a) (b) (c)

Fig. 30. Recognition of walking people from stereo data: Walking motions from the training data are shown in the first four subspace dimensions. Each person is shown with a distinct color and symbol. Small black circles denote the estimated subspace coefficients, α_i, obtained from video of people whose motions were included in the training set. The small black triangles depict subspace coefficients obtained from video of people whose motions were not included in the training set. (a) First two PCA components of a model learned from 4 subjects. Notice that in the first two dimensions the estimated coefficients for the test subject are easily confused with those of the training subjects. (b-c) First four components of a model learned with 9 subjects. In the first four dimensions the motions of the training subjects cluster nicely, and the subspace coefficients estimated for a test subject do not lay close to any one cluster of the training subjects.

is $f = T + 1$.

Figure 30 (a) depicts the first two subspace coefficients, α_i, for the database used for the tracking. The four subjects of the subspace are well separated in the first two dimensions. The estimated coefficients for each one of the two examples depicted by Fig. 25 are shown as circles and a triangle represents the estimated value for the subject in Fig. 26 whose motion is not included in the training dataset. For both women, the first two recovered coefficients fall in the center of the cluster formed by their recorded motion vectors. Also note that while the new subject's motion does appear consistent with one of the training subjects in the first two subspace dimensions, they are quite different in the next two dimensions.

Figure 30 (b,c), depicts the first four subspace coefficients, α_i, for a model learned using nine subjects. The estimated coefficients for each one of the two examples depicted in Fig. 25 are shown as circles and as triangles for the subject of Fig. 26 whose motion is not recorded in the database. Once more, for both women, the first four recovered coefficients fall in the center of the cluster formed by their recorded motion vectors using optical motion capture, meaning that they have been well estimated. Higher order coefficients exhibit small variations that can be attributed to the fact that walking on a treadmill changes the style. Typically the subjects tend to bend the back more when performing the walking in a treadmill to maintain balance. For the man whose motion was not recorded in the database, the recovered coefficients fall within two different clusters when looking at the first two coefficients or at the third and fourth, meaning that this person forms a different cluster

in four dimensions. It is not recognized as any of the nine persons of the database.

The use of motion instead of pose allow us to simply use a closest neighboourd algorithm for classification. Note that if we use pose (see Fig. 5), the recognition is more difficult and a more complex classification algorithm, such as SVM or Adaboost, should be used.

7 Conclusion and Future Work

We have presented an approach to incorporating strong motion models that yields full 3–D reconstruction using a single-hypothesis hill-climbing approach. This results in much lower computational complexity than the current multi-hypothesis techniques. We have demonstrated the effectiveness of our approach for monocular and multi-view tracking of cyclic motions as walking and running and acyclic motions as golf swinging.

The major limitation of the current approach is the number of examples needed to create a database with good generalization properties. We are currently investigating non linear probabilistic techniques that reduces considerably the number of examples required [34].

References

[1] A. Blake, B. North, M. Isard, Learning Multi-Class Dynamics, Advances in Neural Information Processing Systems 11 (1999) 389–395.

[2] N. R. Howe, M. E. Leventon, W. T. Freeman, Bayesian reconstructions of 3D human motion from single-camera video, in: Neural Information Processing Systems, MIT Press, Cambridge, MA, 1999.

[3] V. Pavlovic, J. Rehg, J. MacCormick, Impact of Dynamic Model Learning on Classification of Human Motion, in: Conference on Computer Vision and Pattern Recognition, Vol. 1, 2000, pp. 788–795.

[4] B. North, A. Blake, A. Isard, J. Rittscher, Learning and classification of complex dynamics , IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (9) (2000) 1016–1034.

[5] K. Choo, D. Fleet, People tracking using hybrid monte carlo filtering, in: International Conference on Computer Vision, Vol. 2, Vancouver, Canada, 2001, pp. 321–328.

[6] J. Deutscher, A. Blake, I. Reid, Articulated Body Motion Capture by Annealed Particle Filtering, in: Conference on Computer Vision and Pattern Recognition, Hilton Head Island, SC, 2000, pp. 2126–2133.

[7] M. Isard, A. Blake, Condensation - conditional density propagation for visual tracking, International Journal of Computer Vision 29 (1) (1998) 5–28.

[8] H. Sidenbladh, M. J. Black, D. J. Fleet, Stochastic Tracking of 3D human Figures using 2D Image Motion, in: European Conference on Computer Vision, Vol. 2, 2000, pp. 702–718.

[9] C. Sminchisescu, B. Triggs, Kinematic Jump Processes for Monocular 3D Human Tracking, in: Conference on Computer Vision and Pattern Recognition, Vol. I, Madison, WI, 2003, pp. 69–76.

[10] T. Moeslund, Computer vision-based motion capture of body language, Ph.D. thesis, Aalborg University, Aalborg, Denmark (June 2003).

[11] T. Moeslund, E. Granum, A Survey of Computer Vision-Based Human Motion Capture, Computer Vision and Image Understanding 81 (3) (2001) 231–268.

[12] A. Bottino, A. Laurentinni, A silhouette based technique for the reconstruction of human movement, Computer Vision and Image Understanding 83 (2001) 79–95.

[13] G. Cheung, S.Baker, T. Kanade, Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematis Estimation and Motion Capture, in: Conference on Computer Vision and Pattern Recognition, Madison, WI, 2003, pp. 569–577.

[14] I. Mikic, M. Trivedi, E. Hunter, P. Cosman, Human body model acquisition and tracking using voxel data, International Journal of Computer Vision 53 (3) (2003) 199–223.

[15] Q. Delamarre, O. Faugeras, 3D Articulated Models and Multi-View Tracking with Silhouettes, in: International Conference on Computer Vision, Vol. 2, Corfu, Greece, 1999, pp. 716–721.

[16] T. Drummond, R. Cipolla, Real-time tracking of highly articulated structures in the presence of noisy measurements, in: International Conference on Computer Vision, Vol. 2, Vancouver, Canada, 2001, pp. 315–320.

[17] K. Grauman, G. Shakhnarovich, T. Darrell, Inferring 3D structure with a statistical image-based shape model, in: International Conference on Computer Vision, Nice, France, 2003, pp. 641–648.

[18] G. J. Brostow, I. Essa, D. Steedly, V. Kwatra, Novel Skeletal Representation For Articulated Creatures, in: European Conference on Computer Vision, Vol. 3, Prague, Czech Republic, 2004, pp. 66–78.

[19] C. W. Chu, O. C. Jenkins, M. J. Mataric, Markerless kinematic model capture from volume sequences., in: Conference on Computer Vision and Pattern Recognition, Madison, WI, 2003, pp. 475–482.

[20] A. Agarwal, B. Triggs, 3d human pose from silhouettes by relevance vector regression, in: Conference on Computer Vision and Pattern Recognition, Vol. 2, 2004, pp. 882–888.

[21] A. Elgammal, C. Lee, Inferring 3D Body Pose from Silhouettes using Activity Manifold Learning, in: CVPR, Vol. 2, Washington, DC, 2004, pp. 681–688.

[22] G. Mori, X. Ren, A. Efros, J. Malik, Recovering Human Body Configurations: Combining Segmentation and Recognition, in: Conference on Computer Vision and Pattern Recognition, Vol. 2, Washington, DC, 2004, pp. 326–333.

[23] R. Rosales, S. Sclaroff, Infering Body Pose without Tracking Body Parts, in: Conference on Computer Vision and Pattern Recognition, Vol. 2, 2000, pp. 506–511.

[24] J. Sullivan, S. Carlsson, Recognizing and tracking human action, in: European Conference on Computer Vision, Vol. 1, 2002, pp. 629–644.

[25] A. J. Davison, J. Deutscher, I. D. Reid, Markerless motion capture of complex full-body movement for character animation, in: Eurographics Workshop on Computer Animation and Simulation, Springer-Verlag LNCS, 2001, pp. 3–14.

[26] A. Agarwal, B. Triggs, Tracking articulated motion with piecewise learned dynamical models, in: European Conference on Computer Vision, Vol. 3, Prague, 2004, pp. 54–65.

[27] D. Ormoneit, H. Sidenbladh, M. Black, T. Hastie, Learning and tracking cyclic human motion, in: Advances in Neural Information Processing Systems 13, 2001, pp. 894–900.

[28] L. Herda, R. Urtasun, P. Fua, Hierarchical Implicit Surface Joint Limits for Human Body Tracking, Computer Vision and Image Understanding 99 (2) (2005) 189–209.

[29] C. Sminchisescu, B. Triggs, Covariance Scaled Sampling for Monocular 3D Body Tracking, in: Conference on Computer Vision and Pattern Recognition, Vol. 1, Hawaii, 2001, pp. 447–454.

[30] H. Murase, R. Sakay, Moving object recongition in eigenspace representation: Gait analysis and lip reading., Pattern Recognition Letters 17 (1996) 155–162.

[31] A. Rahimi, B. Recht, T. Darrell, Learning appearance manifolds from video, in: Conference on Computer Vision and Pattern Recognition, San Diego, CA, 2005, pp. 868–875.

[32] C. Sminchisescu, A. Jepson, Generative Modeling for Continuous Non-Linearly Embedded Visual Inference, in: International Conference in Machine Learning, Vol. 69, Banff, Alberta, Canada, 2004, pp. 96–103.

[33] T. Tian, R. Li, S. Sclaroff, Articulated Pose Estimation in a Learned Smooth Space of Feasible Solutions, in: CVPR Learning Workshop, Vol. 3, San Diego, CA, 2005.

[34] R. Urtasun, D. J. Fleet, A. Hertzman, P. Fua, Priors for people tracking from small training sets, in: International Conference on Computer Vision, Beijing, China, 2005, pp. 403–410.

[35] M. Alexa, W. Mueller, Representing animations by principal components, in: Eurographics, Vol. 19, 2000, pp. 411–418.

[36] V. Blanz, C. Basso, T. Poggio, T. Vetter, Reanimating Faces in Images and Video, in: Eurographics, Vol. 22, Granada, Spain, 2003.

[37] M. Brand, A. Hertzmann, Style Machines, Computer Graphics, SIGGRAPH Proceedings (2000) 183–192.

[38] R. Urtasun, P. Glardon, R. Boulic, D. Thalmann, P. Fua, Style-based motion synthesis, Computer Graphics Forum 23 (4) (2004) 799–812.

[39] Y. Yacoob, M. J. Black, Parametric Modeling and Recognition of Activities, in: International Conference on Computer Vision, Mumbai, India, 1998, pp. 120–127.

[40] N. Troje, Decomposing biological motion: A framework for analysis and synthesis of human gait patterns, Journal of Vision 2 (5) (2002) 371–387.

[41] K. Shoemake, Animating Rotation with Quaternion Curves, Computer Graphics, SIGGRAPH Proceedings 19 (1985) 245–254.

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

[43] M. Tipping, C. Bishop, Probabilistic principal component anlaysis, Journal of the Royal Statistical Society, B 61 (3) (1999) 611–622.

[44] W. Press, B. Flannery, S. Teukolsky, W. Vetterling, Numerical Recipes, the Art of Scientific Computing, Cambridge U. Press, Cambridge, MA, 1992.

[45] R. Plänkers, P. Fua, Articulated Soft Objects for Multi-View Shape and Motion Capture, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (9) (2003) 1182–1187.

[46] J. MacCormick, M. Isard, Partitioned sampling, articulated objects, and interface-quality hand tracking, in: European Conference on Computer Vision, Vol. 2, 2000, pp. 3–19.

[47] R. Urtasun, D. J. Fleet, P. Fua, Monocular 3–d tracking of the golf swing, in: Conference on Computer Vision and Pattern Recognition, Vol. 2, San Diego, CA, 2005, pp. 932–938.

[48] A. Jepson, D. J. Fleet, T. El-Maraghi, Robust on-line appearance models for vision tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (10) (2003) 1296–1311.

[49] H. Sidenbladh, M. J. Black, L. Sigal, Implicit Probabilistic Models of Human Motion for Synthesis and Tracking, in: European Conference on Computer Vision, Vol. 1, Copenhagen, Denmark, 2002, pp. 784–800.

[50] A. Agarwal, B. Triggs, Learning to Track 3D Human Motion from Silhouettes, in: International Conference in Machine Learning, Banff, Alberta, Canada, 2004.

[51] V. Lepetit, A. Shahrokni, P. Fua, Robust Data Association For Online Applications, in: Conference on Computer Vision and Pattern Recognition, Vol. 1, Madison, WI, 2003, pp. 281–288.

[52] R. Urtasun, P. Fua, 3D Human Body Tracking using Deterministic Temporal Motion Models, in: European Conference on Computer Vision, Vol. 3, Prague, Czech Republic, 2004, pp. 92–106.

[53] R. Urtasun, P. Fua, Human Motion Models for Characterization and Recognition, in: Automated Face and Gesture Recognition, Seoul, Korea, 2004, pp. 17–22.