From Canonical Poses to 3–D Motion Capture using a Single Camera

Andrea Fossati, Miodrag Dimitrijevic, Vincent Lepetit, Pascal Fua

{Andrea.Fossati,

Miodrag.Dimitrijevic,Vincent.Lepetit,Pascal.Fua}@epfl.ch

Ecole Polytechnique Fédérale de Lausanne (EPFL)

Computer Vision Laboratory, I&C Faculty

CH-1015 Lausanne, Switzerland

http://cvlab.epfl.ch

Abstract

We combine detection and tracking techniques to achieve robust 3–D motion recovery of people seen from arbitrary viewpoints by a single and potentially moving camera. We rely on detecting key postures, which can be done reliably, using a motion model to infer 3–D poses between consecutive detections, and finally refining them over the whole sequence using a generative model.

We demonstrate our approach in the cases of golf motions filmed using a static camera and walking motions acquired using a potentially moving one. We will show that our approach, although monocular, is both metrically accurate because it integrates information over many frames and robust because it can recover from a few misdetections.

Index Terms

Computer vision, Motion, Video Analysis, 3D Scene Analysis, Modeling and recovery of physical attributes, Tracking.

I. Introduction

Recent approaches to modeling people's 3–D motion from video sequences can be roughly classified into those that detect specific postures in individual frames and those that track the motion from frame to frame given an initial pose. The first category usually involves matching against a large image database and is becoming increasingly popular, but requires very large training datasets to be effective. The second category involves predicting the pose in a frame given the pose computed in previous ones, which can easily fail if errors start accumulating in the prediction, causing the estimation process to diverge.

Neither technique is clearly superior to the other, and both are actively investigated. In this paper, we show that they can be combined to accurately reconstruct the 3–D motion of people seen from arbitrary viewpoints using a single, and potentially moving, camera. At the heart of our approach is the fact that human motions often contain characteristic postures that are relatively easy to detect. Given two consecutive such postures, modeling intermediate poses becomes an interpolation problem, which is much easier to solve reliably than open-ended tracking.

More specifically, we show that we can reconstruct 3D golfing motions filmed using a static camera and walking motions acquired using a potentially moving one. In the golf case, the easy-to-detect postures are the starting position, when the golfer transitions from upswing to downswing, and the final one. For walking, they are the ones that occur at the end of each step when people have their legs furthest apart. We therefore use a chamfer-based method [11] that was designed to detect key postures from any viewpoint, even when the background is cluttered and background subtraction is impractical because the camera moves as is the case in the first row of Fig. 1. Because the detected postures are projections of 3–D models, we can map them back to full 3–D poses and use them to select and warp motions from a training database that closely match them. This yields initial pose estimates such as those of the second row of Fig. 1. It lets us create the synthetic images we would see if the person truly were in those positions. These images are depicted by the figure's third row and we refine the pose until they match the real ones. This yields the results depicted by the two last rows of Fig. 1.

The importance of combining detection and tracking to achieve robustness has long been known [9], [23] and manually introducing a few 3–D keyframes in a tracking algorithm has been shown to be effective [10]. More recently, a fully automated approach to combining tracking and

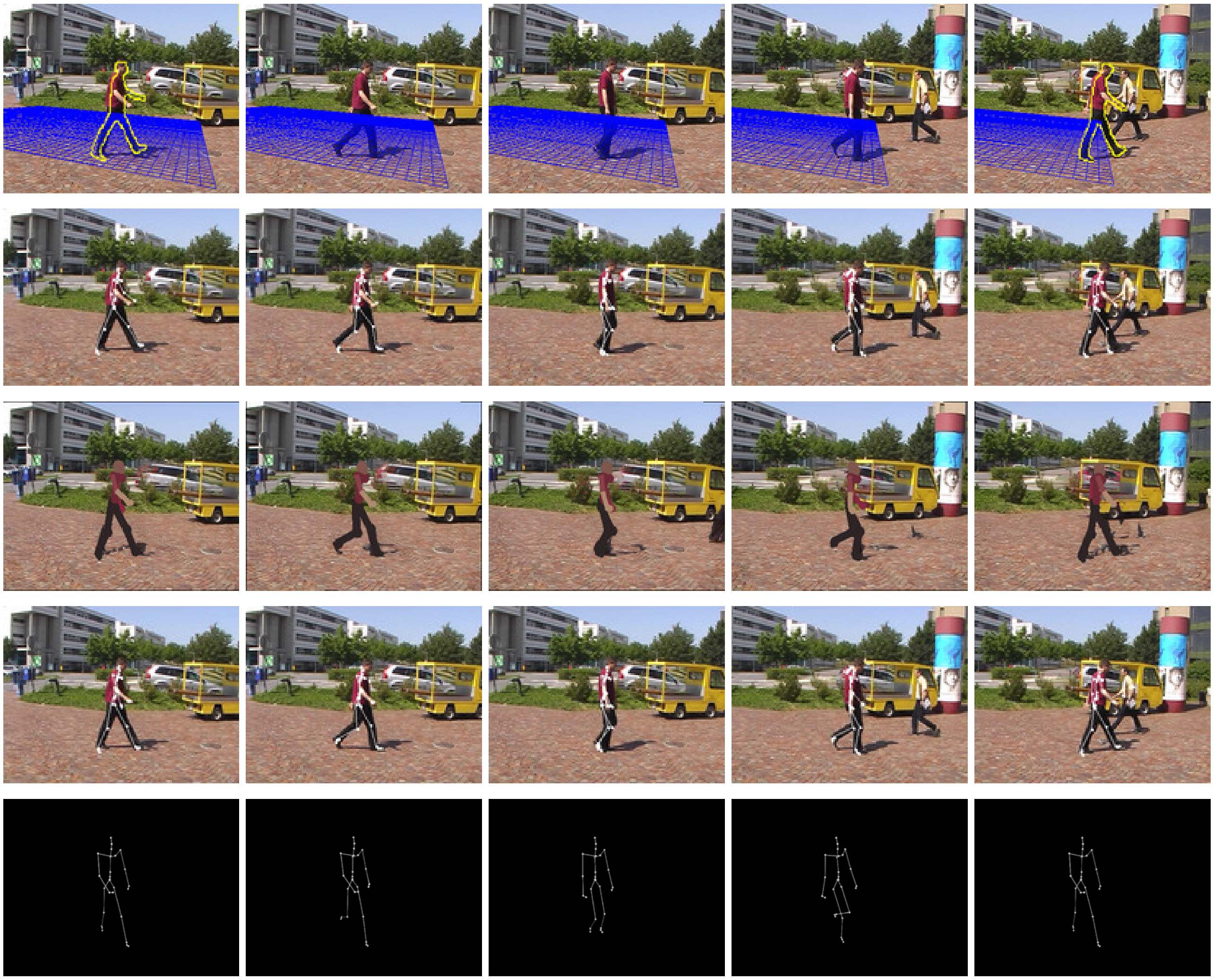


Figure 1. Our approach. **First row:** Input sequence acquired using a moving camera with silhouettes detected at the beginning and the end of the walking cycle. The projection of the ground plane is overlaid as a blue grid. **Second row:** Projections of the 3–D poses inferred from the two detections. **Third row:** Synthesized images that are most similar to the input: Different colors represent different appearence models. **Fourth row:** Projections of the refined 3–D poses. **Fifth row:** 3–D poses seen from a different viewpoint.

detection has been shown to be robust at following multiple people over very long sequences in [28] in 2–D. This is achieved by detecting people in canonical poses and tracking them from there, which still has the potential to diverge. By contrast, interpolating between detected silhouettes prevents this and yields 3–D reconstructions.

We chose walking and golfing to demonstrate our approach because we had access to both the relevant motion databases and silhouette detection techniques. The framework, however, is general because most human motions include very characteristic postures that are easier to detect than completely arbitrary ones. Athletic motions are a good example of this. Canonical postures can be detected when a tennis player hits the ball with a forehand, a backhand, or a serve [38]. In a work environment, there also are very characteristic poses between which people alternate, such as sitting at their desk and walking through doors.

The fact that canonical postures are common is important because one of the limitations of state-of-the-art detection-based approaches to 3–D motion reconstruction is that huge training databases would be required to detect all possible postures. By contrast, if one only needs to detect a few easily recognizable postures, much smaller databases should suffice, thus making the approach easier to deploy.

II. Related Work

Existing approaches to video-based 3–D motion capture remain fairly brittle for many reasons: Humans have a complex articulated geometry overlaid with deformable tissues, skin, and loose clothing. Their motion is often rapid, complex, self-occluding and presents joint reflection ambiguities. Furthermore, the 3–D body pose is only partially recoverable from its projection in one single image, where usually the background is cluttered and the resolution is poor. Reliable and robust 3–D motion analysis therefore requires good tracking across frames, which is difficult because of the poor quality of image-data and frequent occlusions. Recent approaches to handling these problems can roughly be classified into those that

- *Detect:* This implies recognizing postures from a single image by matching it against a database and has become increasingly popular recently [40], [1], [13], [24], [20], [12], [19], [30], [25], [4], [42] but requires very large sets of examples to be effective. Moreover this often relies on background subtraction and on clean silhouettes, such as those that can be extracted from the HumanEva dataset [35], which require static cameras or controlled environments. Finally these methods are usually able just to obtain a good reconstruction of the body pose but cannot correctly locate it in a 3D environment.
- *Track:* This involves predicting the pose in a frame given observation of the previous one. It requires thus an initial pose and can easily fail if errors start accumulating in the prediction, causing the estimation process to diverge. The possibility of drifting is usually mitigated by introducing sophisticated statistical techniques for a more effective search [9], [7], [8],

[45], [46], [37], by using strong dynamic motion models as priors [33], [27], [34], [2], [43], [39], [29] or even by introducing physics-based models [5].

Neither technique has been proved to be superior, and both are actively studied and sometimes combined: Manually introducing a few 3–D keyframes is known to be a powerful way to constrain 3–D tracking algorithms [10], [22]. In the 2–D case, it has recently been shown that this can be done in a fully automated fashion to track multiple people in extremely long sequences [28]. This involves tracking forwards and backwards from individual and automatically detected canonical poses. While effective, this approach to tracking still has the potential to diverge. In this paper, we avoid this problem and go to full 3–D by observing that automated canonical pose detections can be linked into complete trajectories, which let us first recover rough 3–D poses by interpolating between these detections and then refining them by using a generative model over full sequences. A similar approach has been proposed for 3–D hand tracking [41] but makes much stronger assumptions than we do by requiring high-quality images so that the hand outlines can accurately and reliably be extracted from the background.

The work presented in this paper builds on some of our own earlier results. We rely on spatio-temporal templates to detect the people in canonical poses [11] and on PCA-based motion models [43] to perform the interpolation. However, unlike in this latter paper, the system does not require manual initialization. This means that we had to develop a strategy to link detections, infer initial 3–D poses from them, and perform the pose refinement even when the camera moves or the background is cluttered. As a result, we can now operate fully automatically under far more challenging conditions than before.

III. Approach

We first use a template-based approach [11] to detect people in poses that are most characteristic of the target activity, as shown in the first row of Fig. 1. The templates consist of consecutive 2–D silhouettes obtained from 3–D motion capture data seen from six different camera views and at different scales. This way the motion information is incorporated into the templates and helps to distinguish actual people who move in a predictable way from static objects whose outlines roughly resemble those of humans. For each detection, the system returns a corresponding 3–D pose estimate.

In theory, a person should be detected every time a key pose is attained, which the template-based algorithm does very reliably. The few false positives tend to correspond to actual people but detected at somewhat inaccurate scales or orientations and false negatives occur when the relative position of the person with respect to the camera generates an ambiguous projection and the key pose becomes hard to distinguish from others. In our experiments, this almost never happened in the golfing case and sometimes did in the walking case when the camera moved and saw the subject against a cluttered background and from a difficult angle. To handle such cases, we have implemented a Viterbi-style algorithm that links detections into consistent trajectories, even though a few may have been missed. Since the camera may move, we perform this computation in the ground plane, which we relate to the image plane via a homography that is recomputed from frame to frame.

Finally, we use consecutive detections to select and time-warp motions from a training database obtained via optical motion capture. As shown in the second row of Fig. 1, this gives us a rough estimate of the body's position and configuration in each frame between detections. To refine this initial estimate, and since the camera may move from frame to frame, we first compute homographies between consecutive frames and use them to synthesize a background image from which the moving person has been almost completely removed. When we know the camera to be static, we synthesize the background image by simple median filtering of the images between detections. We then learn an appearance model from the detections and use it in conjunction with the synthesized background to produce new images, which lets us refine the body position by minimizing an objective function that represents the dissimilarity between the original and synthetic images. For increased robustness, we perform this minimization over all frames simultaneously. This yields the refined poses depicted by the bottom three rows of Fig. 1.

In the remainder of this section, we first introduce the models we use to represent human bodies and their motion. We then briefly describe our approach first to detecting people in canonical poses, second to using these detections to estimate the motion between frames, and, finally, to refining this estimate.

A. Body and Motion Models

As in [44], we represent the human body as cylinders attached to an articulated 3–D skeleton and use a linear subspace method to represent the motion between canonical poses. This body model has standard dimensions and proportions, thus it allows us to obtain reasonable results on different subjects without the need of being specifically trimmed. Adapting the skeleton proportions would have required an a priori knowledge of their more likely variations, as was done for example in [3] using the SCAPE model. In practice, we have not found it necessary to do so because of the scale ambiguity inherent to monocular reconstruction. Using a model that is slightly too small or too big simply results in variations in the recovered camera position with respect to the subject. A *pose*, whether canonical or not, 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

$$\mathbf{g}_t \in \mathbb{R}^6 \quad . \tag{1}$$

A *motion* between two canonical poses 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 becomes a sequence of N discrete poses

$$\begin{aligned} \Psi &= [\psi_1^T, \cdots, \psi_N^T]^T \in \mathbb{R}^{DN} \quad , \\ \mathbf{G} &= [\mathbf{g}_1^T, \cdots, \mathbf{g}_N^T]^T \in \mathbb{R}^{6N} \quad . \end{aligned} \tag{2}$$

Since motions can occur at different speeds, we encode them at a canonical speed and time-warp them to represent other speeds. We let the pose vary as a function of a phase parameter μ that is defined to be 0 at the beginning of the motion and 1 at the end. For periodic motions such as walking, the phase is periodic. For non-periodic ones such as swinging a golf club, it is not. 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

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

In practice, we learn motion models from optical motion capture data comprising several people performing the same activity several times. For walking, we used a Vicon^{tm} system to capture the motions of four men and four women on a treadmill at speeds ranging from 3 to 7 km/h by increments of 0.5 km/h. 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. We therefore only learn motion models for the joint angles. 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. Similarly, to learn the golf swing model, we asked two golfers to perform 24 swings each.

Because our subjects move at different speeds, 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 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 [31] 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* $\{\Theta_i\}_{i=1}^{m}$:

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

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. In all the experiments shown in this paper, we used $m = 5$, which has proved sufficient to achieve good reconstruction accuracy.

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

$$\psi(\mu, \alpha_1, \cdots, \alpha_m) \;\approx\; \Theta_0(\mu) + \sum_{i=1}^{m} \alpha_i \Theta_i(\mu) \;\; . \tag{5}$$

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

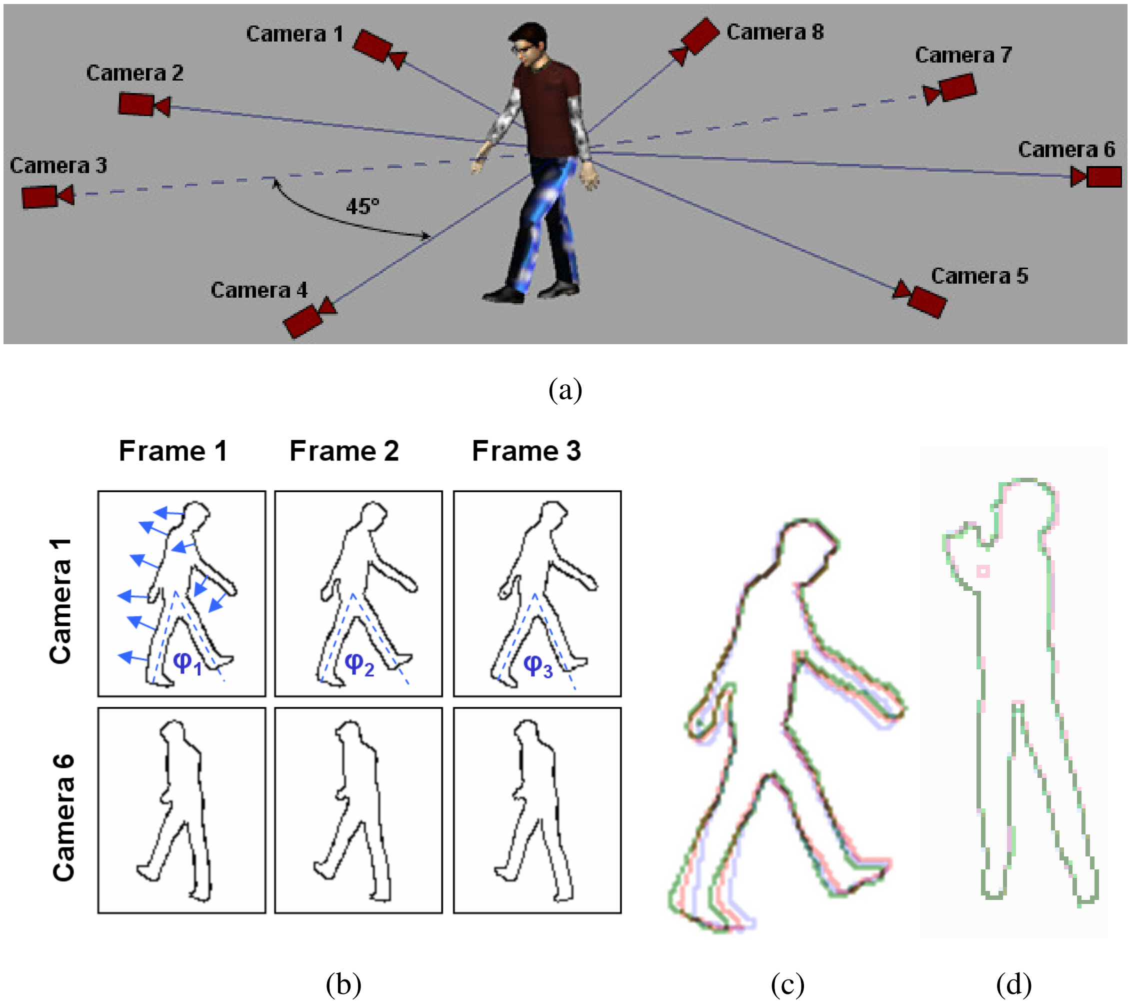


Figure 2. Creating spatio-temporal templates. (a) Six virtual cameras are placed around the model. Viewpoints 3 and 7 are not considered because they are not discriminant enough. (b) A template corresponding to a particular view consists of several silhouettes computed at three consecutive instants. The small blue arrows in image Camera 1 / Frame 1 represent edge orientations used for matching silhouettes for some of the contour pixels. (c) The three silhouettes of a walking template are superposed to highlight the differences between outlines. (d) Superposed silhouettes of a golf swing template.

B. Detection and Initialization

As in our earlier publication [11], people in canonical poses are detected using *spatio-temporal templates* that are sequences of three silhouettes of a person, such as the one of Fig. 2(c). The first corresponds to the moment just before they reach the target pose, the second to the moment when they have precisely the right attitude, and the third just after. Matching these templates against three-image sequences let us differentiate between actual people who move in a predictable way and static objects whose outlines roughly resemble those of humans, which are surprisingly numerous. As a result, it turns out to be much more robust than earlier template-based approaches to people detection [26], [15], [16].

As shown in Fig. 2(a), to build these templates, we introduced a virtual character that can perform the same captured motions we used to build the motion model discussed above and rendered images at a rate of 25 frames per second as seen from virtual cameras in six different orientations. The rendered images are then used to create templates such as those depicted by Fig. 2(b). The rendered images are rescaled at seven different scales ranging from 52×64 to 92×113 pixels, so that an image at one scale is 10% larger than the image one scale below. From each one of the rendered images, we extract the silhouette of the model. Each template is made of the silhouette corresponding to the canonical pose, the one before, and the one after. The silhouettes are represented as sets of oriented pixels that can be efficiently matched against image sequences. We refer the interested reader to our earlier publication for further details [11].

An added bonus of this approach to detecting people, is that to each detection we can associate the set of $\{\alpha_i\}$ PCA coefficients, as defined in Eq. 4. Averaging the coefficients corresponding to two consecutive detections and sampling the μ_n phase parameter of Eq. 3 at regular intervals gives us a pose estimate in each intermediate frame. In the golfing case where the body's center of gravity moves little, this is enough to characterize the whole motion since we can assume that the g_t vector of Eq. 1 that encodes the position and orientation of the body root remains constant except for the component that encodes the rotation around the z axis. In the walking case, this is of course not true and we use the position of the detected silhouettes on the ground plane to estimate the person's 3D location and orientation. We then use spline interpolation to derive initial g_t values in between, as will be discussed in more details in Section IV-C.

C. Refinement

The poses obtained using the method discussed above are only approximative. To refine them, we generate for each one the synthetic images we would see if the person truly were in that pose and compare to the original one. Minimizing the dissimilarity between real and synthetic image then lets us refine the poses in each individual frame.

In our implementation, we depart from typical generative approaches in two important ways. First, to increase robustness, we refine the poses over all the frames simultaneously by optimizing

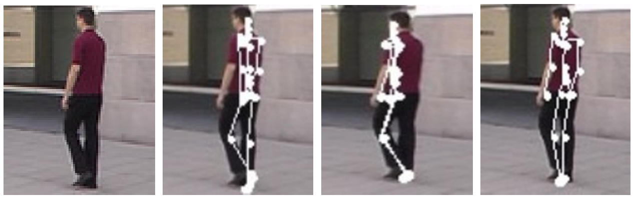


Figure 3. Refinement process. The images are from left to right the input image, the initialization given by the interpolation process, the result obtained without using a background image, and finally the result obtained as proposed. Whole parts of the body can be missed when the background is not exploited.

with respect to the α PCA coefficients of Eq. 4 and $\mathbf{g}_t$ root node positions and orientations of Eq. 1. Second, as shown in the third row of Fig. 1, we not only create an appearance model for the person but also for the background so that the synthetic images we produce include both. As illustrated by Fig. 3, this is important because it effectively constrains the projections of the reconstructed model to be at the right place and allows recovery of the correct pose even when the initial guess is far from it.

To perform the refinement, we define the objective function $L(\hat{\Psi})$ as $-\log(p(\hat{\Psi}|I_1, \ldots, I_N))$ of a pose sequence $\hat{\Psi} = \Psi(\mu_1, \cdots, \mu_N, \mathbf{g}_1, \cdots, \mathbf{g}_N, \alpha_1, \cdots, \alpha_m)$ in an image sequence $I_1, \ldots, I_N$. To compute it we consider the standard Bayesian formula

$$p(\hat{\Psi}|I_1, \ldots, I_N) = \frac{p(I_1, \ldots, I_N|\hat{\Psi}) \cdot p(\hat{\Psi})}{p(I_1, \ldots, I_N)}. \tag{6}$$

The $p(I_1, \ldots, I_N)$ term is constant and can be ignored. Because we have a dependable way to initialize Ψ, we express the prior as a distance from its initial value and write its negative log as

$$-\log p(\hat{\Psi}) = \sum_{k=1}^{m} \left(\frac{\alpha_k - \alpha_k^0}{\sqrt{\lambda_k}} \right)^2, \tag{7}$$

where α_k^0 represents the initialization value for the k^{th} PCA parameter, given by the detections, and λ_k is the eigenvalue associated to the k^{th} eigenvector.

Assuming conditional independence of the appearance in consecutive frames given the motion model, we can decompose $p(I_1, \ldots, I_N|\hat{\Psi})$ as

$$p(I_1, \ldots, I_N|\hat{\Psi}) = \prod_{i=1}^{N} p(I_i|\hat{\psi}_i, \hat{\mathbf{g}}_i), \tag{8}$$

where $\hat{\psi}_i = \psi(\mu_i, \alpha_1, \ldots, \alpha_m)$ is the pose in image I_i, as defined by Eq. 5.

Assuming for simplicity that, given an estimated pose $\hat{\psi}_i$, a background, and a foreground model, all the pixels (u, v) in image I_i are conditionally independent, we write

$$p(I_i|\hat{\psi}_i, \hat{\mathbf{g}}_i) = \prod_{(u,v)\in I_i} p(I_i(u,v)|\hat{\psi}_i, \hat{\mathbf{g}}_i). \tag{9}$$

We estimate $p(I_i(u,v)|\hat{\psi}_i)$ for each pixel of frame i as follows: Given the generated background images, we project our human body model according to pose $\hat{\psi}_i$. As discussed above, individual limbs are modeled as cylinders to which we associate a color histogram obtained from the projected area of the limb in the frames where the silhouettes were detected. We project the body model onto the generated background image to obtain a synthetic image, such as those depicted by the third row of Fig. 1. If (u, v) is located within the projection of a body part, we take $p(I_i(u,v)|\hat{\psi}_i)$ to be proportional to the value of its corresponding bin in the color histogram of the body part. If, instead, (u, v) is located on the background, we take $p(I_i(u,v)|\hat{\psi})$ to be a Gaussian distribution centered on the corresponding pixel value in the synthetic background image B_i, with fixed covariance Σ. We therefore write

$$p(I_i(u,v)|\hat{\psi}, \hat{\mathbf{g}}) = \begin{cases} h_{part(u,v)}(I_i(u,v)) & \text{if } (u,v) \in F \\ N(B_i(u,v), \Sigma; I_i(u,v)) & \text{if } (u,v) \notin F \end{cases},$$

where F represents the projection of the body model into the image. Modeling both the foreground and background appearence helps in achieving more accuracy and robustness, as already noted in the literature [18], [32] for static camera cases.

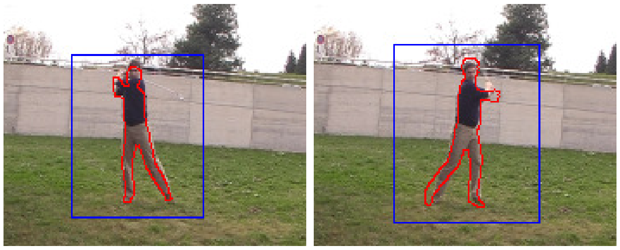
Given Eqs. 7, 8 and 9, we can write

$$L(\hat{\Psi}) = -\log(p(\hat{\Psi})) + \sum_{i=1}^{N} \sum_{(u,v)\in I_i} -\log(p(I_i(u,v)|\hat{\psi}_i)) \tag{10}$$

and refine all the poses between detections by minimizing $L(\hat{\Psi})$ with respect to $(\mu_1, \cdots, \mu_N, \mathbf{g}_1, \cdots, \mathbf{g}_N, \alpha_1, \cdots, \alpha_m)$, which define the motion in the whole sequence. This minimization is performed stochastically by sampling particles thrown in the parameter space around the initialization.

IV. From Detections to Trajectories

To reconstruct golf swings, we treat as canonical poses the transition between the upswing and the downswing and the end of the upswing, as shown in Fig. 4. Since there is a little



(a) (b)

Figure 4. Key pose detections at the beginning and at the end of a golf swing.

motion of the golfer's center of gravity during the swing, we take the $\mathbf{g}_t$ vectors of Eq. 1 to be initially all equal, and during the optimization we only allow a rotation around the z axis. Furthermore, since the camera is static in the examples we use, we simply median filter frames in the whole sequence to synthetize the background image we need to perform the pose refinement of Section IV-C.

To track walking people, we use the beginning of the walking cycle, when the legs are furthest apart, as our canonical pose. Our spatio-temporal templates detect this pose reliably but with the occasionnal false positive and false negative. Such errors must be eliminated and the valid detections linked into consistent trajectories, which is more involved than in the golf case since people move over time and the camera must be allowed to move also to keep them in view.

In this section, we first describe the linking procedure, discuss how we initialize the $\mathbf{g}_t$ vectors that encode the person's global motion, and, finally, account for the fact that the direction in which people face and in which they move are strongly correlated.

A. Linking Detections

In our scheme, people should be detected at the beginning of every walking cycle but are occasionally missed. To link these sparse detections into a complete trajectory, we have implemented a Viterbi-style algorithm. It is important to note that the system is trained on a very specific pose, the left leg in front of the right one, which helps our algorithm resolve ambiguities by giving higher scores to the correct detections. We could have used two keyposes instead of one for each walking cycle, but we empirically found that using one was a good trade-off between tracking robustness and computational load. As shown in Section V even missing a detection out of two can still lead to reliable results. Finally these detections include not only an image

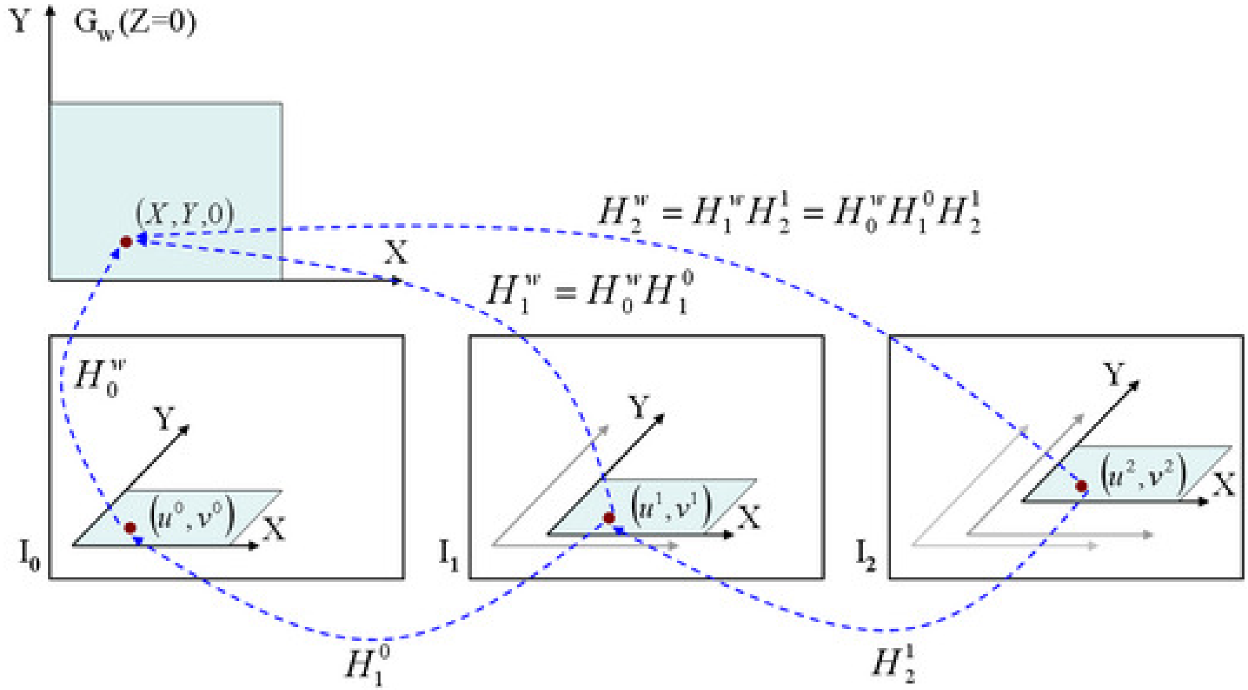


Figure 5. Ground plane tracking. Given the corresponding interest points in each pair of the consecutive frames in the sequence $I_i, i = 0..N$ it is possible to compute the homographies between these frames $H_{i+1}^i, i = 0..N-1$. Further, knowing the initial homography between the world ground plane and the reference image H_0^w, we compute the required homographies between each of the frames and the world ground plane $H_i^w, i = 1..N$.

location but also the direction the person faces, which is an important clue for linking purposes.

a) Ground Plane Registration: Since the camera may move, we work in the ground plane, which we relate to each frame by a homography that is computed using a standard technique [36], which is illustrated in Fig. 5. In practice, we manually indicate the ground plane in one frame and compute an initial homography between it and the world ground plane G_w – H_0^w. Then, we detect interest points in both the reference frame and the next one and match them. From the set of correspondences we compute the homography between the subsequent frame ground plane and the reference frame ground plane H_1^0, and further from the subsequent frame ground plane and the world ground plane H_1^w. We repeat this process for all the frames $I_i, i = 1..N$ obtaining the homographies between them and world ground plane $H_i^w, i = 1..N$, N being the number of the sequence frames. This makes it easy to compute the world ground plane coordinates of the detections knowing the 2D coordinates in the frame ground plane. Since there are specific

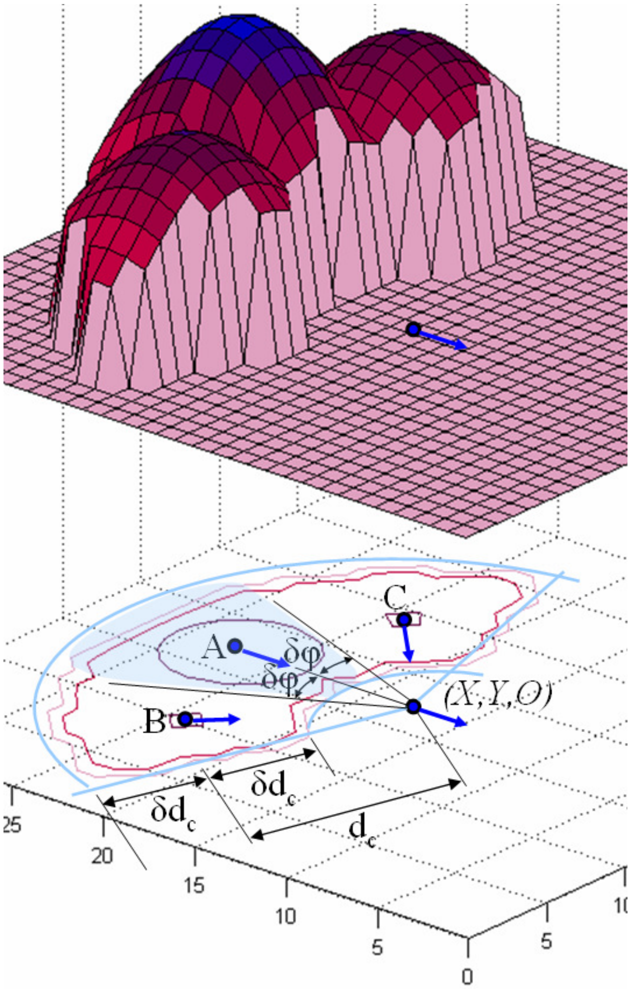


Figure 6. Transitional probabilities for hidden state (X, Y, O). They are represented by three Gaussian distributions corresponding to three possible previous orientations. Each Gaussian covers a 2D area bounded by two circles of radii $d_c - \delta d_c$ and $d_c + \delta d_c$, where δd_c represents an allowable deviation from the mean, and by two lines defined by tolerance angle $\delta\varphi$.

orientations associated with each detection, we also recalculate these orientations with respect to the world ground plane.

b) Formalizing the Problem: The homographies let us compute ground plane locations and one of the possible orientations for all detections, which we then need to link while ignoring potential misdetections. To this end, we define a hidden state at time t as the oriented position of a person on the ground plane $L_t = (X, Y, O)$, where t is a frame index, (X, Y) are discretized ground plane coordinates, and O is one of the possible orientations.

We introduce the maximum likelihood estimate of a person's trajectory ending up at state i

at time t

$$\Gamma_t(i) = \max_{l_1,...,l_n} P(I_1, L_1 = l_1, ..., I_n, L_n = l_n) \ \ , \tag{11}$$

where I_j represents the j^{th} frame of the video sequence. Casting the computation of Γ in a dynamic programming framework requires introducing probabilities of observing a particular image given a state and of transitioning from one state to the next.

We therefore take b_{it}, the probability of observing frame I_t given hidden state i, to be

$$b_{it} = P(I_t | L_t = i) \sim \frac{1}{d_{\text{bayes-chamfer}}} \ \ , \tag{12}$$

where $d_{\text{bayes-chamfer}}$ is a weighted average of the chamfer distances between projected template contours and actual image edges. This makes sense because the coefficients used to weight the contributions are designed to account for the relevance of different silhouette portions in a Bayesian framework [11].

We also introduce the probability of transition from state j at time t' to state i at time t

$$a_{ji}^{\Delta t} = P(L_t = i | L_{t'} = j), \text{with } \Delta t = t - t'. \tag{13}$$

Since we only detect people when their legs are spread furthest apart, we can only expect a detection approximately every $N_c = 30$ frames for an average v = 5 km/h walking speed in a 25 Hz video. This implies an average distance $d_c = \frac{vN_c}{25}$ between detections. We therefore assume that $a_{ji}^{\Delta t}$ for state $i = (X, Y, O)$ follows a Gaussian distribution centered at (X_μ, Y_μ) such that

$$\sqrt{(X - X_\mu)^2 + (Y - Y_\mu)^2} = d_c \ \ , \tag{14}$$

and positioned in the direction 180° opposite to the orientation O, as depicted by point A in Fig. 6. This Gaussian covers only the hidden states with orientation equal to O. The other previous states from which a transition may occur are those with orientations $O + \pi/4$ and $O - \pi/4$, which are covered by two neighboring Gaussians, as depicted by points B and C in Fig. 6.

c) Linking Sparse Detections: Given the probabilities of Eq. 12 and 13, if we could expect a detection in every frame, linking them into complete trajectories could be done using the Viterbi algorithm to recursively maximize the Γ_t maximum likelihood of Eq. 11.

However, since we can only expect a detection approximately every $N_c = 30$ frames, we allow the model to change state directly from $L_{t'}$ at time t' to L_t at time t $(t' < t)$, $N_c - \delta t < t - t' < N_c + \delta t$ and skip all frames in between. δt is a frame distance tolerance that we set to 10 in our implementation.

This lets us reformulate the maximization problem of Eq. 11 as one of maximizing

$$\begin{aligned} \Gamma_t(i) &= \max_{l_{t_1},\ldots,l_{t_n}} P(I_{t_1}, L_{t_1} = l_{t_1}, ..., I_{t_n}, L_{t_n} = l_{t_n}) \ , \\ &= b_{it} \max_{j,\Delta t}(a_{ji}^{\Delta t}\Gamma_{t-\Delta t}(j)) \ , \end{aligned} \tag{15}$$

where $t_1 < t_2 < ... < t_n$, n are the indices of frames I in which at least one detection occurred and $N_c - \delta t < \Delta t < N_c + \delta t$.

This formulation lets us initially retain for each detection several hypotheses with different orientations and allow the dynamic programming algorithm to select those that provide the most likely trajectories according to the probabilities of Eq. 12 and 13. If a detection is missing, the algorithm simply bridges the gap using the transition probabilities only. For the sequences of Fig. 7 and Fig. 9, this yields the results depicted by Fig. 8 and Fig. 10.

B. Predicting 3D Poses between Detections

A complete trajectory computed as discussed above includes rough estimates of the body's position, orientation, and 3D pose parameterized by a set of joint angles for the frames in which the key posture was detected. We use straightforward spline interpolation to predict positions and orientations of the body between the two detections, which allows us to initialize the g vectors of Eq. 1.

The whole procedure is very simple and naturally extends to the case where a key posture has been missed, which can be easily detected by comparing the number of frames between consecutive detections and the median value for the whole sequence. In this case, a longer motion must be created by concatenating several motion cycles —usually 2, and never more than 3 in our experiments— depending on the number of frames between detections. This new motion is then resampled as before. Obviously the initial predictions then lose in accuracy, but they usually remain precise enough to retrieve the correct poses thanks to the refinement process described in the following subsection.

C. Refining the Predicted Poses

To track a person walking about, the camera usually has to move to keep him in view. Therefore we cannot use a simple background subtraction technique to create the background image we require for refinement purposes and adopt the more sophisticated approach depicted

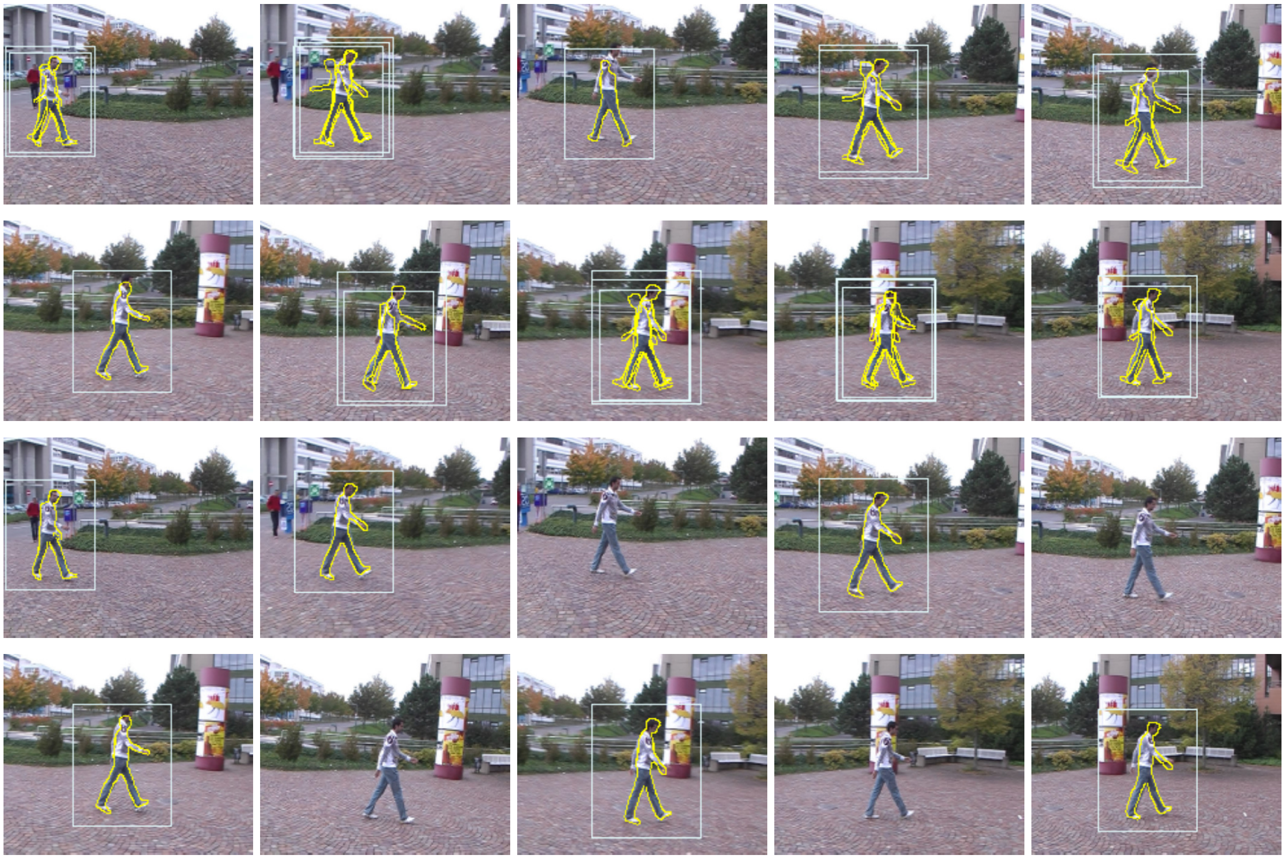


Figure 7. Filtering silhouettes with temporal consistency on an outdoor sequence acquired by a moving camera. **First two rows:** Detection hypotheses. **Third and fourth row:** Detections after filtering out the detection hypotheses that do not lie on the recovered most probable trajectory. Note that the extremely similar poses in which is very hard to distinguish which leg is in front of which leg, are successfully disambiguated by our algorithm.

by Fig. 11. We treat each image of the sequence in turn as a reference and consider the few images immediately before and after. We compute homographies between the reference and all other images [17], [36], which is a reasonable approximation of the frame-to-frame deformation because the time elapsed between successive frames is short and lets us warp all the images into the reference frame. Then, by computing the median of the values for each pixel in HSV color space, we obtain background images with few artifacts.

To account for the fact that walking trajectories are smooth both spatially and temporally, we do not treat the $\mathbf{g}_i$ and μ_i as independent from each other. Instead, as we did for initialization purposes, we represent trajectories as 2–D splines lying on the ground plane and whose shape is completely defined by the position and orientation of the body root node at the endpoints of

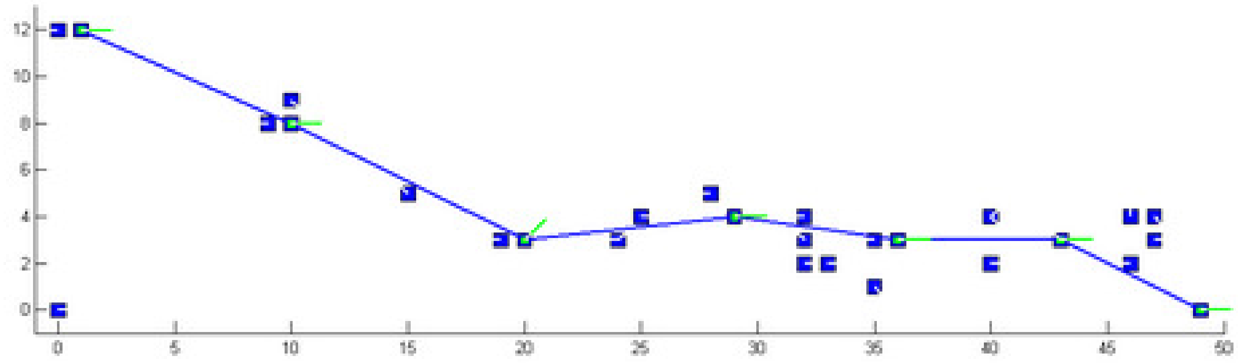


Figure 8. Recovered trajectory for the sequence depicted by Fig. 7. Dark blue squares represent detection hypotheses and bright short lines inside them represent the detection orientations. Smaller light green squares and lines represent the retained detections and their orientations respectively. These detections form the most probable trajectory depicted by dark blue lines.

a sequence, which we denote as $\mathbf{g}_{start}$ and $\mathbf{g}_{end}$. In other words, we write all the $\mathbf{g}_i$ as functions of $\mathbf{g}_{start}$ and $\mathbf{g}_{end}$. Similarly, we introduce a parameter $0 < \mu_c < 1$ that defines what percentage of the walking cycle has been accomplished during the first half of the sequence and derive all the other μ_i by simple interpolation. If the speed remains constant during a walking cycle, the value of μ_c is 0.5. In practice, it can go from 0.3 to 0.7 if the person speeds-up or slows-down between the first and the second half-cycle.

We can now refine the pose between two detections by optimizing the objective function $L(\hat{\Psi})$ of Eq. 10 with respect to $(\mu_c, \alpha_1, \ldots, \alpha_m, \mathbf{g}_{start}, \mathbf{g}_{end})$. Our experiments have shown that using this reduced set of variables regularizes the motion and yields much better convergence properties than using the full parameterization. However, this formulation does not exploit the fact that people usually walk in the direction they are facing, which means that the body global position, which is controlled by the first three variables of the 6–D $\mathbf{g}_{start}$ and $\mathbf{g}_{end}$ vectors is not independent from the other three, which control orientation. We can therefore further improve our results by adding an additional term to our objective function to enforce this constraint. We define

$$L_{walk}(\hat{\Psi}) = L(\hat{\Psi}) + \beta \sum_{i=2}^{N} (\phi^2_{(i-1)\to i}) \tag{16}$$

where $\phi_{(i-1)\to i}$ is the angle between the direction the person faces and the direction of motion

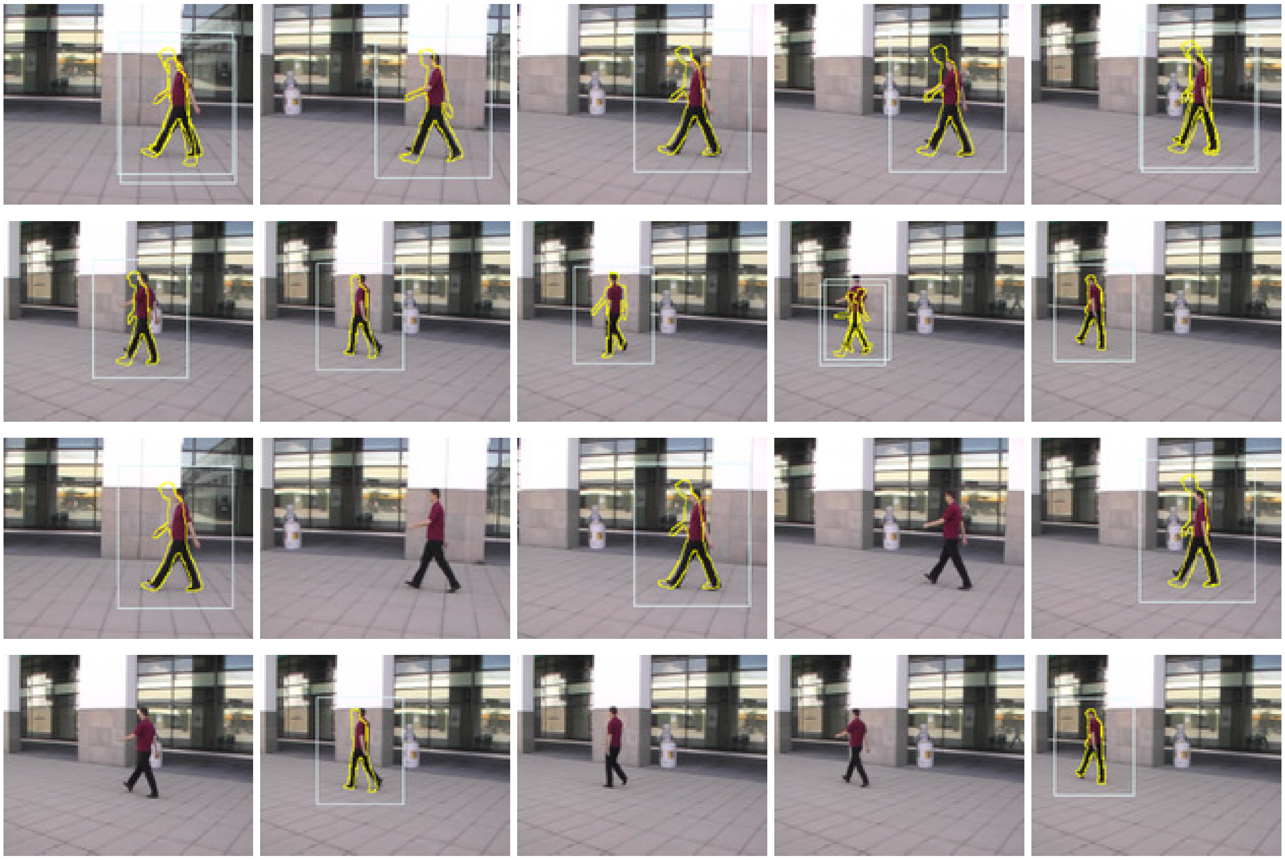


Figure 9. Filtering silhouettes with temporal consistency on an outdoor sequence acquired by a moving camera. **First two rows:** Detection hypotheses. **Third and fourth row:** Detections after filtering out the detection hypotheses that do not lie on the recovered most probable trajectory.

and β is a weighting term which is kept constant for all our experiments, and whose purpose is to make the two terms of the same order of magnitude. As demonstrated in [14] and as will be shown in Section V, minimizing $L_{walk}(\hat{\Psi})$ instead of $L(\hat{\Psi})$ has little influence on the recovered poses but yields more realistic global body orientations.

V. Results

In this section, we present our results on golfing and walking sequences that feature subjects *other* than those we used to create our motion databases and seen from many different perspectives. A computationally expensive part of the algorithm is the refinement step of Section IV C since, for each particle, we must render the whole sequence, be it a walking cycle or a golf swing, and compute the image likelihood for each frame. Therefore finding the best solution

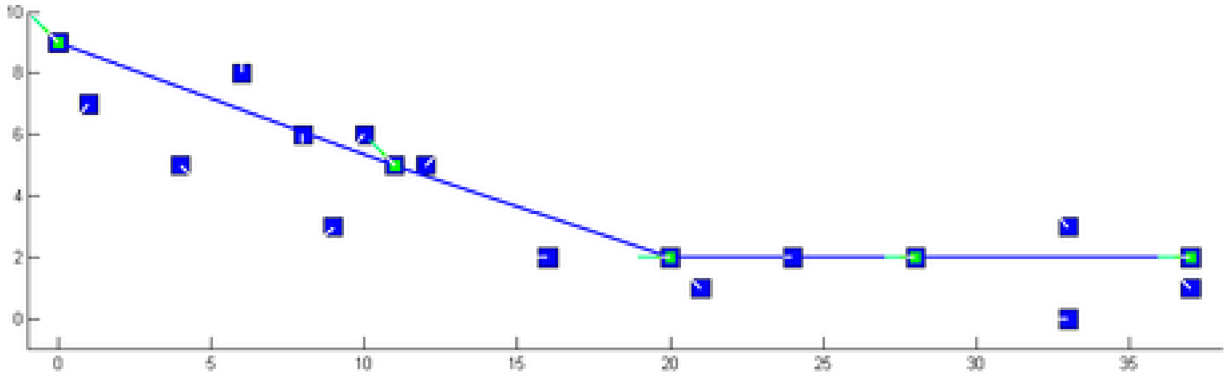


Figure 10. Recovered hypothetical trajectory for the sequence depicted by Fig. 9. Dark blue squares represent detection hypotheses and bright short lines inside them represent the detection orientations. Smaller light green squares and lines represent the retained detections and their orientations respectively. These detections form the most probable trajectory depicted by dark blue lines.

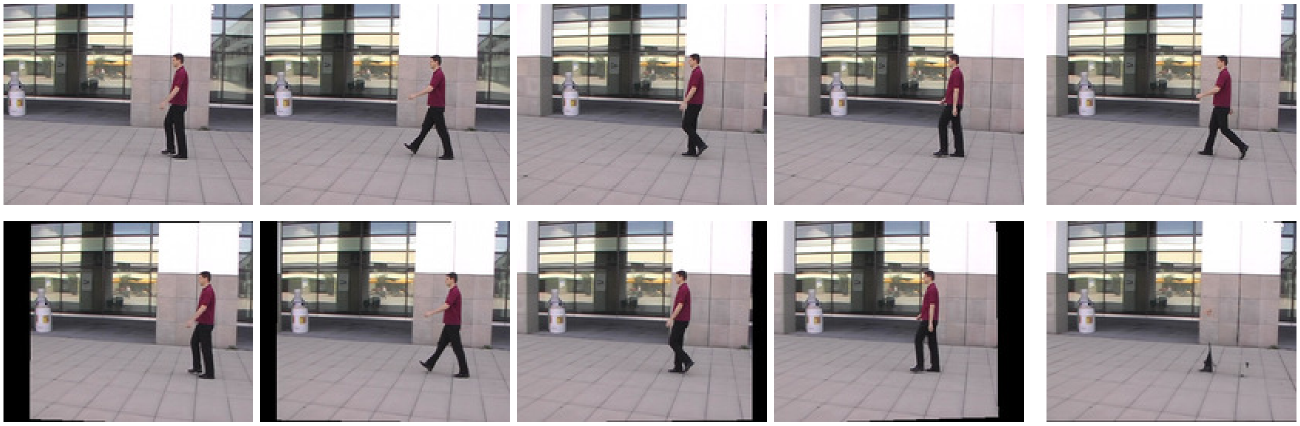


Figure 11. Synthesizing a background image. **First row:** The rightmost image is the reference image whose background we want to synthesize. The other 4 are those before and after it in the sequence. **Second row:** The same four images warped to match the reference image. Computing the median image of these and the reference image yields the rightmost image, which is the desired background image.

for an activity can require around 10 minutes on a standard computer, in our current MATLAB implementation.

A. Golfing

Fig. 12 depicts a golf swing by a professional golfer. By contrast, Fig. 13 depicts one performed by one of the authors of this paper who does not play golf and whose motion is therefore far from

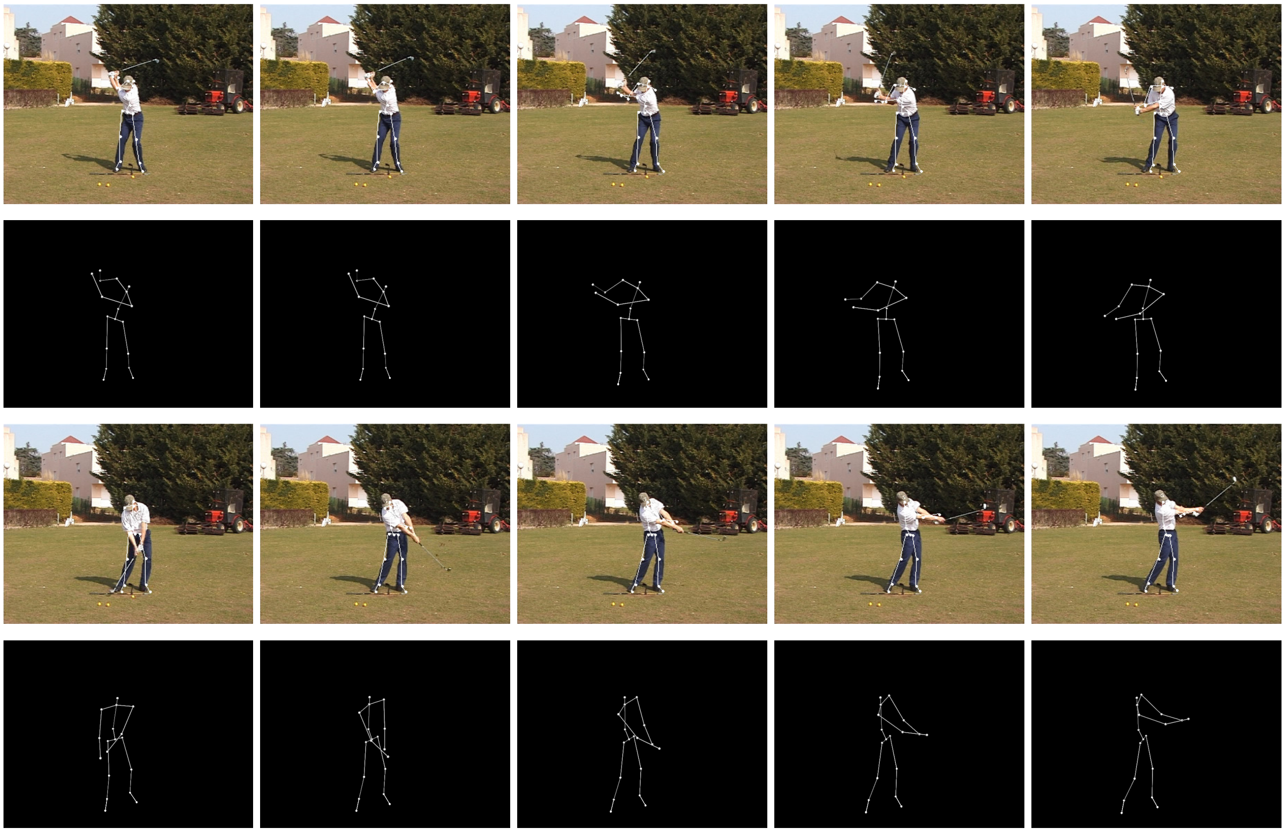


Figure 12. Reconstructing a golf swing performed by a professional player. **First and third row:** Frames from the input video with reprojected 3D skeletons. **Second and fourth row:** 3D skeleton seen from a different viewpoint. The corresponding videos are given as supplementary material.

correct. In both cases, our system correctly detects the key postures and recovers a 3D trajectory without any human intervention. This demonstrates that it is robust not only to the relatively low quality of the imagery but also to potentially large variations in the exact motion being recovered. Fig. 14 shows the background model that was recovered and used to generate the results of Fig. 12. Note that the feet are mistakenly made part of the background reconstruction and this results in unwarranted motion of the feet. This is easily fixed by constraining them to remain on the ground, as show in the supplementary material.

B. Walking

We now demonstrate the performance of our algorithm on walking sequences acquired under common but challenging conditions. In all cases except when we use the HumanEva dataset [35] to quantify our results, the subject is seen against a cluttered background and the camera moves

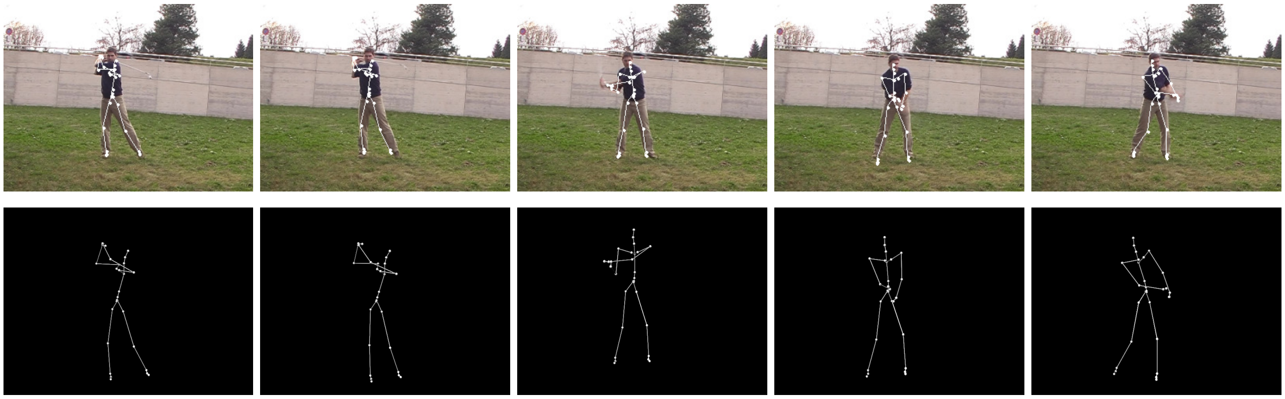


Figure 13. Reconstructing a golf swing performed by a novice player. **First row:** Frames from the input video with reprojected 3D skeletons. **Second row:** 3D skeleton seen from a different a different viewpoint.

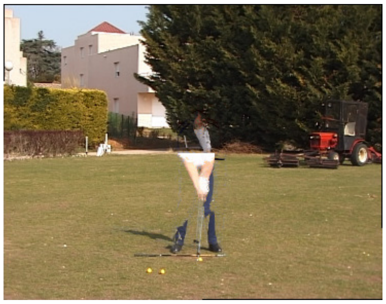


Figure 14. Background image used to generate the results of Fig. 12. Notice that there are some artifacts for instance in the feet area, which are anyway overcome by our algorithm.

to follow him, which precludes the use of simple background subtraction techniques.

In the sequences of Figs. 15 and 16 the camera translates. Furthermore, in Fig. 16, the subject is seen first from the side and progressively from the back as he becomes smaller and smaller. In the sequence of Fig. 17, the subject walks along a circular trajectory and the camera follows him from its center. At some point the subject undergoes a total occlusion but the global model

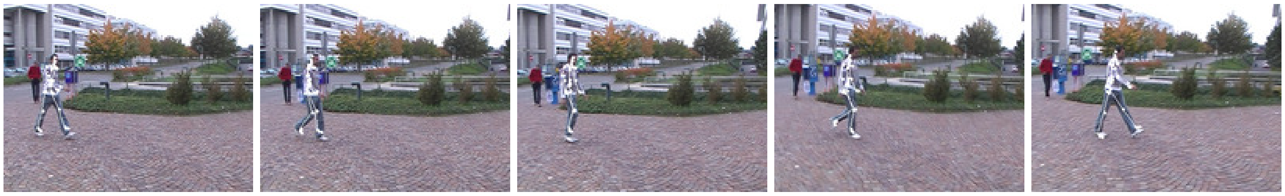


Figure 15. Recovered 3D skeletons reprojected into individual images of the sequence of Fig. 7, which was acquired by a camera translating to follow the subject.

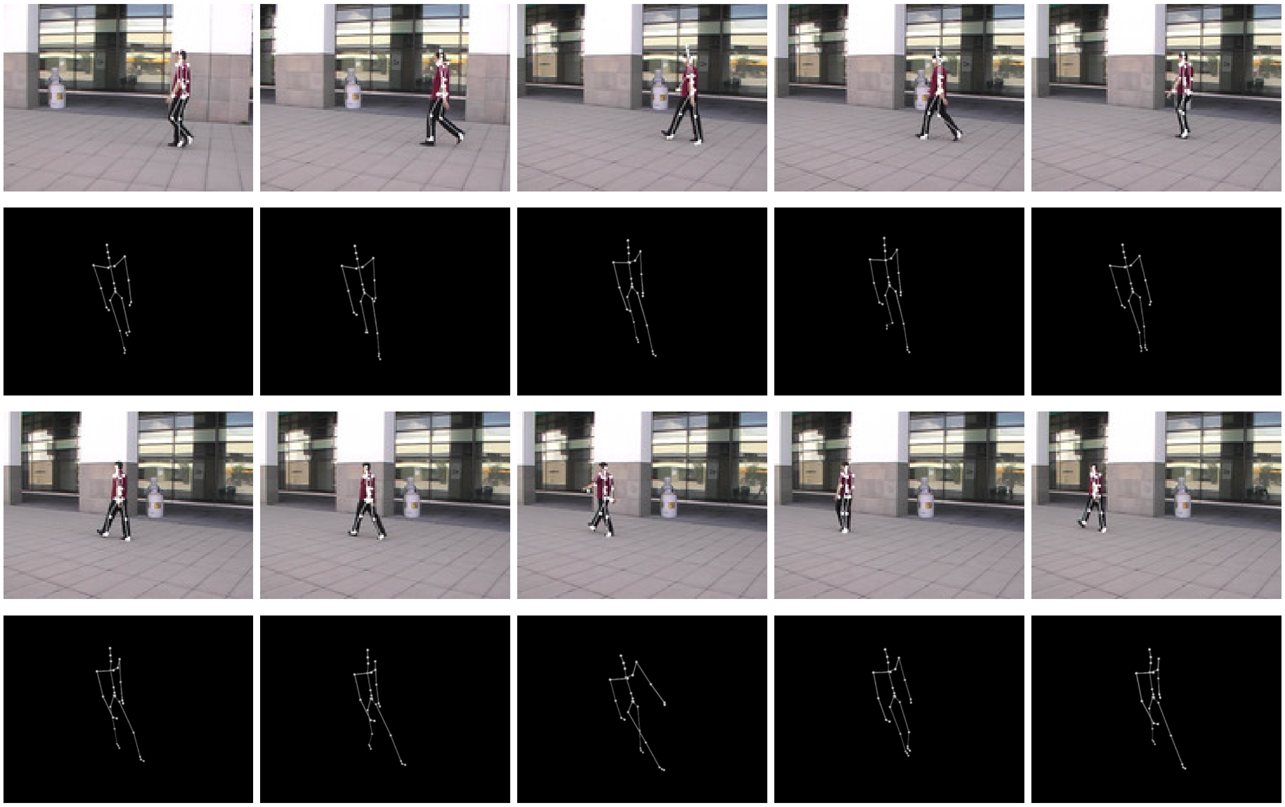


Figure 16. Final result for the subject of Fig. 9 who moves away from the camera and is eventually seen from behind. **First and third rows:** Frames from the input video with reprojected 3D skeletons. **Second row and fourth rows:** 3D skeletons seen from a different viewpoint. The 3–D pose is correctly estimated over the sequence, even when the person goes far away and eventually turns his back to the camera. Note that the extremely similar poses in which it is very hard to distinguish which leg is in front are successfully disambiguated by our algorithm.

allows the algorithm to nevertheless recover both pose and position for the whole sequence. We can also recover the instantaneous speeds and the ground plane trajectory, as shown in Fig. 18.

All these results were obtained by minimizing the objective function of Eq. 16 that explicitly enforces consistency between the direction the person faces and the direction of motion. We also computed results by minimizing the objective function of Eq. 10, which does not take this consistency into account. When shown in projections in the original images, these two sets of results are almost indistinguishable. However, the improvement becomes clear when one compares the two trajectories of Fig. 18, one obtained without enforcing the constraint and the other with. To validate these results, we manually marked the subject's feet every 10 frames in the sequence of Fig. 17 and used their position with respect to the tiles on the ground plane to estimate their 3D coordinates. We then treated the vector joining the feet as an estimate of

	X Error		Y Error		Orientation Error	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Not Linking Orientation to Motion	12.0	7.1	16.8	11.9	11.7	7.6
Linking orientation to Motion	11.8	7.3	14.9	9.3	6.2	4.9

Table I

COMPARING THE RECOVERED POSITION AND ORIENTATION VALUES FOR THE BODY ROOT NODE AGAINST GROUND TRUTH DATA FOR THE SEQUENCE OF FIG. 17. WE PROVIDE THE MEAN AND STANDARD DEVIATION OF THE ABSOLUTE POSITIONAL ERROR IN THE X AND Y COORDINATES, IN CENTIMETERS, AND THE MEAN AND STANDARD DEVIATION OF THE RECOVERED ORIENTATION ERROR, IN DEGREES.

the body orientation and the midpoint as an estimate of its location. As can be seen in Table I, linking orientation to motion produces a small improvement in the position estimate and a much more substantial one in the orientation estimate, which is consistent with what can be observed in Fig. 18. Obviously these numbers should be only considered in a relative way, and to have an idea of the quantitative performance of our algorithm we refer the reader to the results on the HumanEvaII sequence.

In the sequence of Fig. 19 the subject walks along a curvilinear path and the camera follows him, so that the viewpoint undergoes large variations. We are nevertheless able to recover pose and motion in a consistent way, as shown in Fig. 20 that depicts the recovered trajectory. Again, linking orientation to motion yields improved results.

Fig. 21 demonstrates the robustness of our approach to missed detections. We ran our algorithm on the same sequence as in Fig. 1 but ignored one out of every two detections. Note that, even though the subject is now only detected every other step, the algorithm's performance barely degrades.

To further quantify our results, we tracked subject S4 of the HumanEvaII dataset [35] over 230 frames acquired by camera C1. Since it is static, we used the same simple approach as in the golf case to synthesize the background image we use to compute our image likelihoods. In Fig. 22 we plot the mean 3–D distance between the real position of some reference points and those recovered by our algorithm, which are commensurate with the numerical results of Table I that we obtained using our own sequences. Given that our approach is strictly monocular—we simply ignored the input of the other cameras—the 158mm average error our algorithm produces

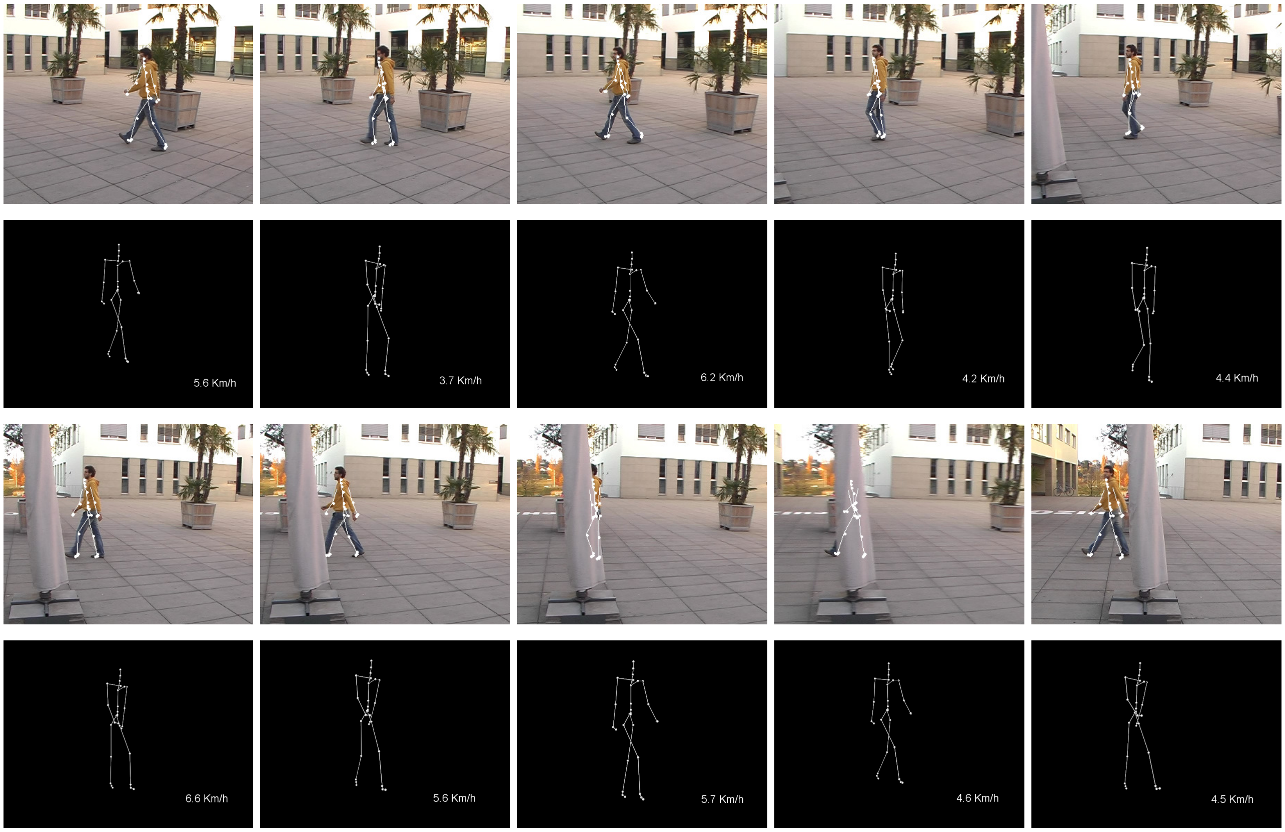
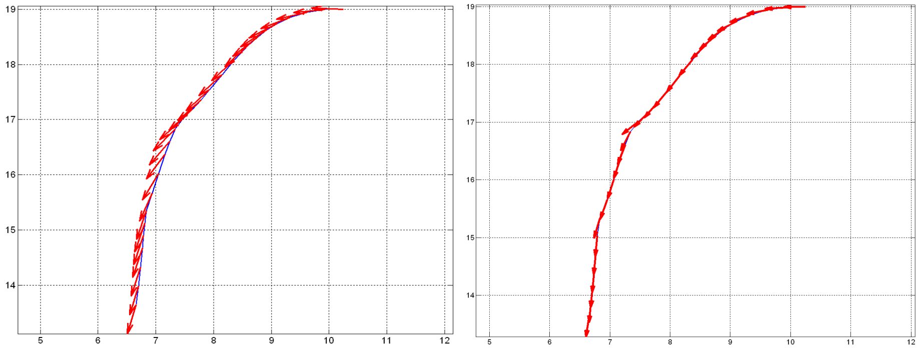


Figure 17. Subject walking in a circle. **First and third rows:** Frames from the input video with reprojected 3D skeletons. **Second and fourth rows:** 3D skeletons seen from a different viewpoint. The numbers in the bottom right corner are the instantaneous speeds derived from the recovered motion parameters. The corresponding videos are submitted as supplementary material.



(a) (b)

Figure 18. Recovered 2D trajectory of the subject of Fig. 17. The underlying grid is made of 1×1 meter squares and the arrows represent the direction he is facing. (a) When orientation and motion are not linked, he appears to walk sideways. (b) When they are, he walks naturally.

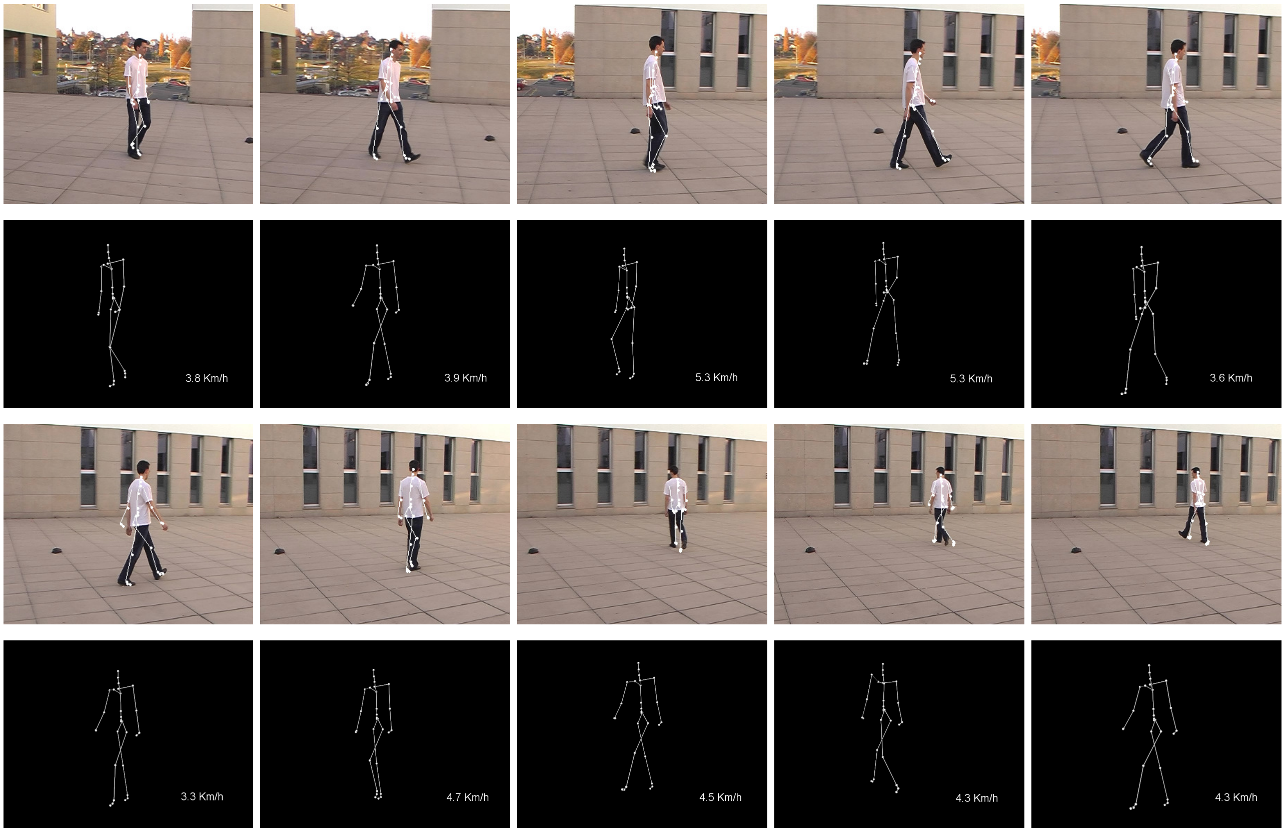
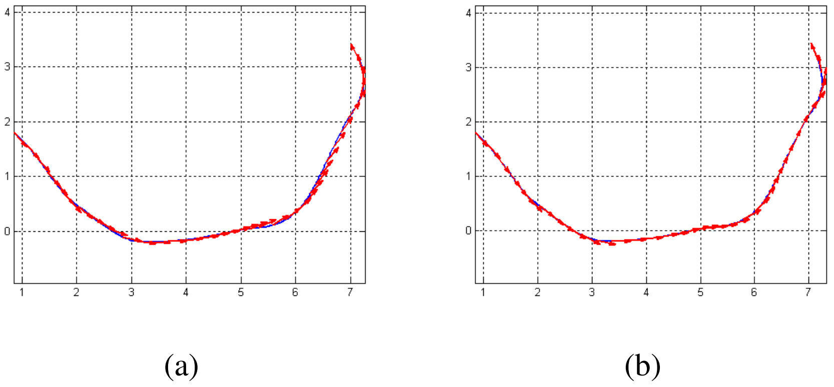


Figure 19. Pedestrian tracking and reprojected 3D model in a second sequence. **First and third rows:** Frames from the input video with reprojected 3D skeletons. **Second and fourth rows:** 3D skeletons seen from a different viewpoint. The numbers in the bottom right corner are the instantaneous speeds derived from the recovered motion parameters.



(a)

(b)

Figure 20. Recovered 2D trajectory of the subject of Fig. 19. As in Fig. 18, when orientation and motion are not linked, he appears to walk sideway (a) but not when they are (b).

is within the range of methods that make similar assumptions. By comparison, errors around 200mm are reported in [21] and between 100 and 200mm in [6]. This is encouraging given the

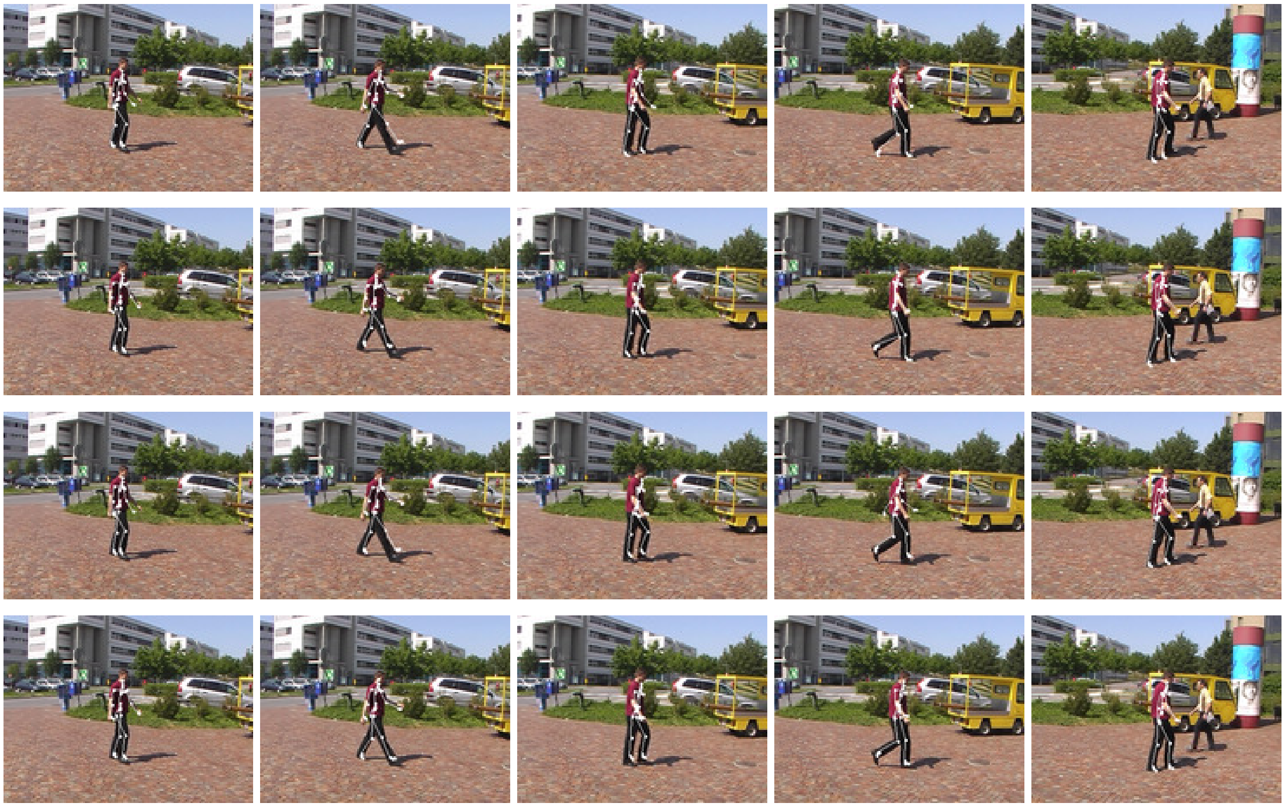


Figure 21. Robustness to misdetection. **First two rows:** Initial and refined poses for a sequence in which 3 consecutive key-poses are detected. **Last two rows:** Initial and refined poses for the same sequence when ignoring the central detection and using the other two. The initial poses are less accurate but the refined ones are indistinguishable.

fact that we only use relatively coarse models and motions described by a reduced number of parameters. In other words, our algorithm is designed more for robustness, moving cameras, and recovery from situations where other algorithms might lose track, such as total occlusions, than for accuracy.

Of course the algorithm, even if it is designed for robustness, can fail. In the walking case, this can happen if the subject performs very sharp turns, thus preventing the Viterbi algorithm to infer the correct trajectory. Similarly, facing the camera for too long can result in loss of track since our detector is designed for people not seen completely frontally. This could be overcome by adding an appropriate detector, which would be fairly easy to do since it could take advantage of the very reliable frontal head detection algorithms that now exist. In the golfing case, a misdetection of either the initial or the final pose would also cause a failure, but they are infrequent because the pose is so characteristic.

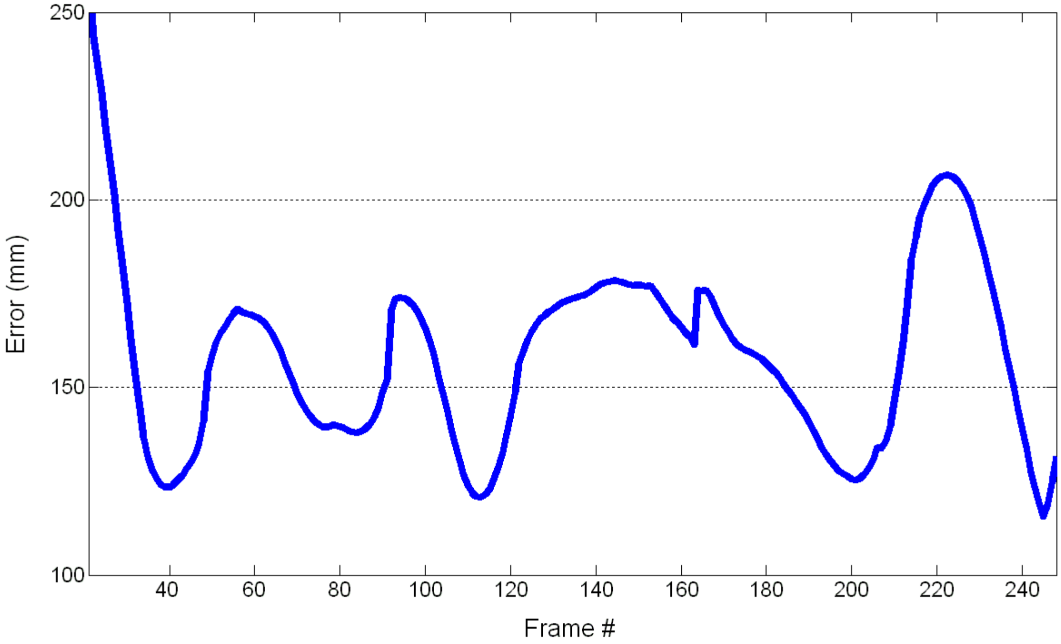


Figure 22. Absolute mean 3D error in joint location obtained on frames 21-248 of the HumanEvaII dataset for subject S4 and using only camera C1 as input. It is expressed in millimeters.

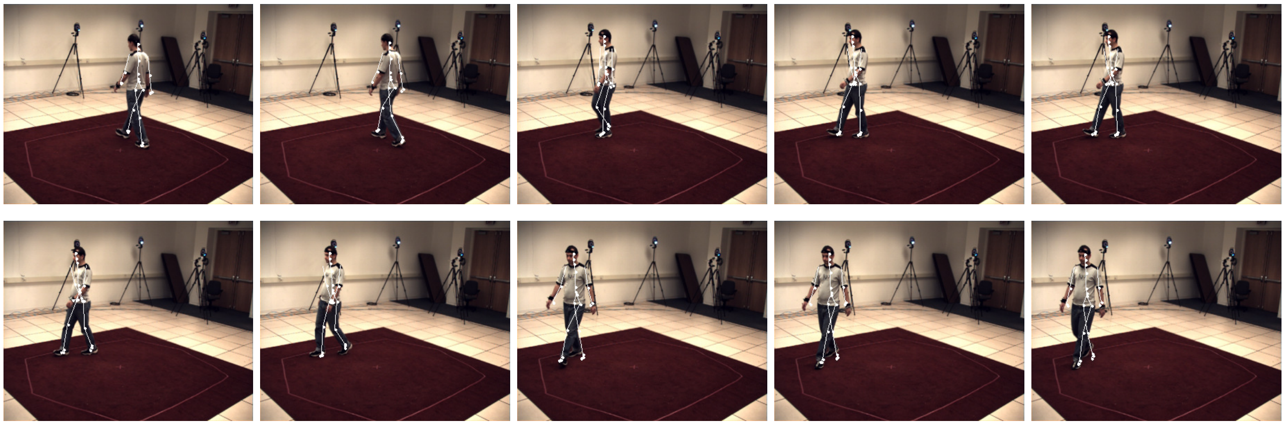


Figure 23. Tracking subject S4 from the HumanEvaII dataset using only camera C1. Obtained results projected onto the input frames.

VI. Conclusion

The walking and golfing motions contain characteristic postures that are relatively easy to detect. We have exploited this fact to formulate 3–D motion recovery from a single video sequence as an interpolation problem. This is much easier to achieve than open-ended tracking and we have shown that it can be solved using straightforward minimization.

This approach is generic because most human motions also feature canonical poses that can be easily detected. This is significant because it means that we can focus our future efforts on developing methods to reliably detect these canonical poses instead of all poses, which is much harder.

A limitation of our approach is that we do not handle transitions from one activity to another, as Markovian motion models could. However, since transitions typically also involve keyposes, the approach could potentially be extended to this much more demanding context given a sufficiently rich training database. This would involve choosing which motion model to use to connect these keyposes and modeling the transition probabilities between activities, and is a topic for future research.

REFERENCES

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *Conference on Computer Vision and Pattern Recognition*, 2004.

[2] A. Agarwal and B. Triggs. Tracking articulated motion with piecewise learned dynamical models. In *European Conference on Computer Vision*, Prague, May 2004.

[3] A.O. Balan and M.J. Black. The naked truth: Estimating body shape under clothing. In *European Conference on Computer Vision*, pages II: 15–29, 2008.

[4] Liefeng Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas. Fast algorithms for large scale conditional 3d prediction. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[5] M. Brubaker, D. Fleet, and A. Hertzmann. Physics-based person tracking using simplified lower-body dynamics. In *CVPR*, Minneapolis, MI, June 2007.

[6] M. Brubaker, A. Hertzmann, and D. Fleet. Physics-based human pose trackng. In *NIPS Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM)*, 2006.

[7] K. Choo and D.J. Fleet. People tracking using hybrid monte carlo filtering. In *International Conference on Computer Vision*, Vancouver, Canada, July 2001.

[8] A. J. Davison, J. Deutscher, and 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.

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

[10] D.E. DiFranco, T.J. Cham, and J.M. Rehg. Reconstruction of 3-D Figure Motion from 2-D Correspondences. In *Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, December 2001.

[11] M. Dimitrijevic, V. Lepetit, and P. Fua. Human Body Pose Detection Using Bayesian Spatio-Temporal Templates. *Computer Vision and Image Understanding*, 104(2-3):127–139, 2006.

[12] E.-J-Ong, A. S. Micilotta, R. Bowden, and A. Hilton. Viewpoint invariant exemplar-based 3–d human tracking. *Computer Vision and Image Understanding*, 104(2–3):178–189, 2006.

[13] A. Elgammal and C.S. Lee. Inferring 3D Body Pose from Silhouettes using Activity Manifold Learning. In *Conference on Computer Vision and Pattern Recognition*, Washington, DC, June 2004.

[14] A. Fossati and P. Fua. Linking pose and motion. In *European Conference on Computer Vision*, Marseille, France, October 2008.

[15] D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *International Conference on Computer Vision*, pages 87–93, 1999.

[16] J. Giebel, D.M. Gavrila, and C. Schnorr. A bayesian framework for multi-cue 3d object tracking. In *Proceedings of European Conference on Computer Vision*, 2004.

[17] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[18] M. Isard and J. MacCormick. Bramble: a bayesian multiple-blob tracker. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 34–41, July 2001.

[19] C.S. Lee and A. Elgammal. Body pose tracking from uncalibrated camera using supervised manifold learning. In *NIPS Workshop on EHuM*, 2006.

[20] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Conference on Computer Vision and Pattern Recognition*, volume 1, San Diego, CA, June 2005.

[21] R. Li, M. Yang, S. Sclaroff, and T. Tian. Evaluation of 3D Human Motion Tracking with a Coordinated Mixture of Factor Analyzers. In *NIPS Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM)*, 2006.

[22] G. Loy, M. Eriksson, J. Sullivan, and S. Carlsson. Monocular 3d reconstruction of human motion in long action sequences. In *European Conference on Computer Vision*, 2004.

[23] K. Mikolajczyk, , R. Choudhury, and C. Schmid. Face detection in a video sequence – a temporal approach. In *Conference on Computer Vision and Pattern Recognition*, 2001.

[24] G. Mori, X. Ren, A.A. Efros, and J. Malik. Recovering Human Body Configurations: Combining Segmentation and Recognition. In *Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2004.

[25] R. Navaratnam, A. Fitzgibbon, and R. Cipolla. The Joint Manifold Model for Semi-supervised Multi-valued Regression. In *International Conference on Computer Vision*, Rio, Brazil, October 2007.

[26] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6:103–113, January 1997.

[27] D. Ormoneit, H. Sidenbladh, M.J. Black, and T. Hastie. Learning and tracking cyclic human motion. In *Neural Information Processing Systems*, pages 894–900, 2001.

[28] D. Ramanan, A. Forsyth, and A. Zisserman. Tracking People by Learning their Appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[29] B. Rosenhahn, T. Brox, and H.P. Seidel. Scaled motion dynamics for markerless motion capture. In *CVPR*, Minneapolis, MI, 2007.

[30] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *International Conference on Computer Vision*, Nice, France, 2003.

[31] K. Shoemake. Animating Rotation with Quaternion Curves. *ACM SIGGRAPH*, 19:245–254, 1985.

[32] H. Sidenbladh and M. J. Black. Learning the statistics of people in images and video. *IJCV*, 54:54–1, 2003.

[33] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic Tracking of 3D human Figures using 2D Image Motion. In *European Conference on Computer Vision*, June 2000.

[34] H. Sidenbladh, M. J. Black, and L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *European Conference on Computer Vision*, Copenhagen, Denmark, May 2002.

[35] L. Sigal and M. J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical report, Department of Computer Science, Brown University, 2006.

[36] G. Simon, A. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *International Symposium on Mixed and Augmented Reality*, pages 120–128, October 2000.

[37] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative Density Propagation for 3–D Human Motion Estimation. In *Conference on Computer Vision and Pattern Recognition*, San Diego, CA, June 2005.

[38] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *European Conference on Computer Vision*, 2002.

[39] L. Taycher, G. Shakhnarovich, D. Demirdjian, and T. Darrell. Conditional Random People: Tracking Humans with CRFs and Grid Filters. In *Conference on Computer Vision and Pattern Recognition*, 2006.

[40] A. Thayananthan, B. Stenger, P.H.S. Torr, and R. Cipolla. Tracking Articulated Hand Motion using a Kinematic Prior. In *British Machine Vision Conference*, pages 589–598, Norwich, UK, 2003.

[41] C. Tomasi, S. Petrov, and A. Sastry. 3d tracking = classification + interpolation. In *International Conference on Computer Vision*, pages 1441–1448, 2003.

[42] R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[43] R. Urtasun, D. Fleet, and P. Fua. 3D People Tracking with Gaussian Process Dynamical Models. In *Conference on Computer Vision and Pattern Recognition*, New York, 2006.

[44] R. Urtasun, D. Fleet, and P. Fua. Temporal Motion Models for Monocular and Multiview 3–D Human Body Tracking. *Computer Vision and Image Understanding*, 104(2-3):157–177, 2006.

[45] Q. Wang, G. Xu, and H. Ai. Learning object intrinsic structure for robust visual tracking. In *Conference on Computer Vision and Pattern Recognition*, Madison, WI, June 2003.

[46] Y. Wu, G. Hua, and T. Yu. Tracking articulated body by dynamic markov network. In *International Conference on Computer Vision*, 2003.