Spatio-temporal Matching for Human Detection in Video

Feng Zhou and Fernando De la Torre

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213. USA

Abstract. Detection and tracking humans in videos have been long-standing problems in computer vision. Most successful approaches (*e.g.*, deformable parts models) heavily rely on discriminative models to build appearance detectors for body joints and generative models to constrain possible body configurations (*e.g.*, trees). While these 2D models have been successfully applied to images (and with less success to videos), a major challenge is to generalize these models to cope with camera views. In order to achieve view-invariance, these 2D models typically require a large amount of training data across views that is difficult to gather and time-consuming to label. Unlike existing 2D models, this paper formulates the problem of human detection in videos as spatio-temporal matching (STM) between a 3D motion capture model and trajectories in videos. Our algorithm estimates the camera view and selects a subset of tracked trajectories that matches the motion of the 3D model. The STM is efficiently solved with linear programming, and it is robust to tracking mismatches, occlusions and outliers. To the best of our knowledge this is the first paper that solves the correspondence between video and 3D motion capture data for human pose detection. Experiments on the Human3.6M and Berkeley MHAD databases illustrate the benefits of our method over state-of-the-art approaches.

1 Introduction

Human pose detection and tracking in videos have received significant attention in the last few years due to the success of Kinect cameras and applications in human computer interaction (*e.g.*, [1]), surveillance (*e.g.*, [2]) and marker-less motion capture (*e.g.*, [3]). While there have been successful methods that estimate 2D body pose from a single image [4–8], detecting and tracking body configurations in unconstrained video is still a challenging problem. The main challenges stem from the large variability of people's clothes, articulated motions, occlusions, outliers and changes in illumination. More importantly, existing extensions of 2D methods [4, 5] cannot cope with large pose changes due to camera view change. A common strategy to make these 2D models view-invariant is to gather and label human poses across all possible viewpoints. However, this is impractical, time consuming, and it is unclear how the space of 3D poses can be uniformly sampled. To address these issues, this paper proposes to formulate the problem of human body detection and tracking as one of spatio-temporal

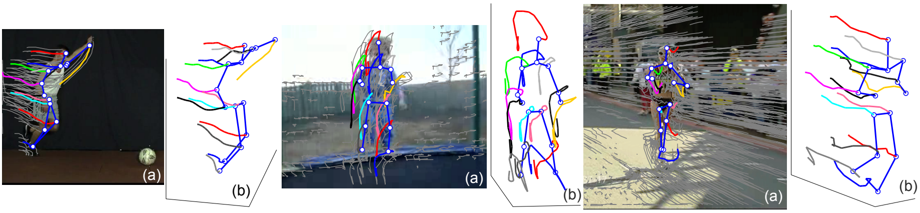

Fig. 1. Detection and tracking of humans in three videos using spatio-temporal matching (STM). STM extracts trajectories in video (gray lines) and selects a subset of trajectories (a) that match with the 3D motion capture model (b) learned from the CMU motion capture data set. Better viewed in color.

matching (STM) between 3D models and video. Our method solves for the correspondence between a 3D motion capture model and trajectories in video. The main idea of our approach is illustrated in Fig. 1.

Our STM algorithm has two main components: (1) a spatio-temporal motion capture model that can model the configuration of several 3D joints for a variety of actions, and (2) an efficient algorithm that solves the correspondence between image trajectories and the 3D spatio-temporal motion capture model. Fig. 1 illustrates examples of how we can rotate our motion capture data model to match the trajectories of humans in video across several views. Moreover, our method selects a subset of trajectories that corresponds to 3D joints in the motion capture data model (about $2-4\%$ of the trajectories are selected). As we will illustrate with the Human3.6M database [9] and the Berkeley MHAD database [10], the main advantage of our approach is that it is able to cope with large variations in viewpoint and speed of the action. This property stems from the fact that we use 3D models.

2 Related work

Early methods for detecting articulated human body in video sequences built upon on simple appearance models with kinematic constraints [11]. State-of-the-art methods for pose detection and body tracking make use of deformable part models (*e.g.* [4, 12, 5, 6]) or regressors [7]. Andriluka*et al.* [13] combined the initial estimate of the human pose across frames in a tracking-by-detection framework. Sapp *et al.* [14] coupled locations of body joints within and across frames from an ensemble of tractable sub-models. Burgos *et al.* [15] merged multiple independent pose estimates across space and time using a non-maximum suppression. More recently, Tian *et al.* [16] explored the generalization of deformable part models [4] from 2D images to 3D spatio-temporal volumes for action detection in video. Zuffi *et al.* [17] exploited optical flow by integrating image evidence across frames to improve pose inference. Compared to previous methods, this paper enforces temporal consistency by matching video trajectories to a spatio-temporal 3D model.

Our method is also related to the work on 3D human pose estimation. Conventional methods rely on discriminative techniques that learn mappings from image features (*e.g.*, silhouettes [18]) to 3D pose with different priors [19, 20]. However, many of them require an accurate image segmentation to extract shape features or precise initialization to achieve good performance in the optimization. Inspired by recent advances in 2D human pose estimation, current works focus on retrieving 3D poses from 2D body part positions estimated by the off-the-shelf detectors [4, 12, 5]. For instance, Sigal and Black [21] learned a mixture of experts model to infer 3D poses conditioned on 2D poses. Simo-Serra *et al.* [22] retrieved 3D poses from the output of 2D body part detectors by a robust sampling strategy. Ionescu *et al.* [7] reconstructed 3D human pose by inferring over multiple human localization hypotheses on images. Inspired by [23], Yu *et al.* [24] recently combined human action detection and a deformable part model to estimate 3D poses. Compared to our approach, however, these methods typically require large training sets to model the large variability of appearance of different people and viewpoints.

3 Spatio-temporal matching

This section describes the proposed STM algorithm. The STM algorithm has three main components: (1) In training, STM learns a bilinear spatio-temporal 3D model from motion capture data, (2) Given an input video, STM extracts 2D feature trajectories and evaluates the pseudo-likelihood of each pixel belonging to different body parts; (3) During testing STM finds a subset of trajectories that correspond to 3D joints in the spatio-temporal model, and compute the extrinsic camera parameters.

3.1 Trajectory-based video representation

In order to generate candidate positions for human body parts, we used a trajectory-based representation of the input video. To be robust to large camera motion and viewpoint changes, we extracted trajectories from short video segments. The input video is temporally split into overlapped video segments of length n frames (*e.g.*, $n = 15$ in all our experiments).

For each video segment, we used [25] to extract trajectories by densely sampling feature points in the first frame and track them using a dense optical flow algorithm [26]. The output of the tracker for each video segment is a set of m_p

trajectories (see notation[1]),

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1^1 & \cdots & \mathbf{p}_{m_p}^1 \\ \vdots & \ddots & \vdots \\ \mathbf{p}_1^n & \cdots & \mathbf{p}_{m_p}^n \end{bmatrix} \in \mathbb{R}^{2n \times m_p},$$

where each $\mathbf{p}_j^i \in \mathbb{R}^2$ denotes the 2D coordinates of the j^{th} trajectory in the i^{th} frame. Notice that the number of trajectories (m_p) can be different between segments. Fig. 2b illustrates a video segment with densely extracted feature trajectories. Compared to the sparser KLT-based trackers [27, 28], densely tracking the feature points guarantees a good coverage of foreground motion and improves the quality of the trajectories in the presence of fast irregular motions.

To evaluate a pseudo-likelihood of each trajectory belonging to a 3D joint, we applied a state-of-the-art body part detector [5] independently on each frame. We selected a subset of $m_q = 14$ body joints (Fig. 2a) that are common across several datasets including the PARSE human body model [5], CMU [29], Berkeley MHAD [10] and Human3.6M [9] motion capture datasets.

For each joint $c = 1 \cdots m_q$ in the i^{th} frame, we computed the SVM score a_{cj}^i for each trajectory $j = 1 \cdots m_p$ by performing an efficient two-pass dynamic programming inference [30]. Fig. 2c shows the response maps associated with four different joints. The head can be easily detected, while other joints are more ambiguous. Given a video segment containing m_p trajectories, we then computed a trajectory response matrix, $\mathbf{A} \in \mathbb{R}^{m_q \times m_p}$, whose element $a_{cj} = \sum_{i=1}^n a_{cj}^i$ encodes the cumulative cost of assigning the j^{th} trajectory to the c^{th} joint over the n frames.

3.2 Learning spatio-temporal bilinear bases

There exists a large body of work that addresses the representation of time-varying spatial data in several computer vision problems (*e.g.*, non-rigid structure from motion, face animation), see [31]. Common models include learning linear basis vectors independently for each frame [32] or discrete cosine transform bases independently for each joint trajectory [33]. Despite its simplicity, using a shape basis or a trajectory basis independently fails to exploit spatio-temporal regularities. To have a low-dimensional model that exploits correlations in space and time, we parameterize the 3D joints in motion capture data using a bilinear spatio-temporal model [34].

[1] Bold capital letters denote a matrix $\mathbf{X}$, bold lower-case letters a column vector $\mathbf{x}$. All non-bold letters represent scalars. $\mathbf{x}_i$ represents the i^{th} column of the matrix $\mathbf{X}$. x_{ij} denotes the scalar in the i^{th} row and j^{th} column of the matrix $\mathbf{X}$. $[\mathbf{X}_1; \cdots; \mathbf{X}_n]$ and $[\searrow_i \mathbf{X}_i]$ denote vertical and diagonal concatenation of sub-matrices $\mathbf{X}_i$ respectively. $\mathbf{1}_{m \times n}, \mathbf{0}_{m \times n} \in \mathbb{R}^{m \times n}$ are matrices of ones and zeros. $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is an identity matrix. $\|\mathbf{X}\|_p = \sqrt[p]{\sum |x_{ij}|^p}$ and $\|\mathbf{X}\|_F = \sqrt{\operatorname{tr}(\mathbf{X}^T\mathbf{X})}$ designate the p-norm and Frobenius norm of $\mathbf{X}$ respectively. $\mathbf{X}^\dagger$ denotes the Moore-Penrose pseudo-inverse.

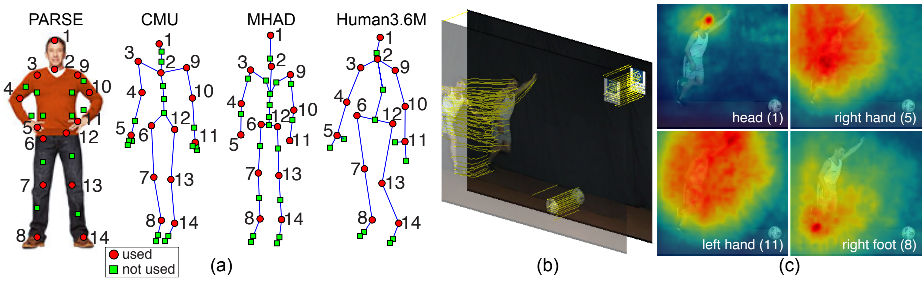

Fig. 2. Example of feature trajectories and their responses. (a) Geometrical configuration of 14 body joints shared across 3D datasets. (b) Dense trajectories extracted from a video segment. (c) Feature response maps for 4 joints (see bottom-right corner).

Given a set of 3D motion capture sequences of different lengths, we randomly select a large number (> 200) of temporal segments of the same length, where each segment denoted by $\mathbf{Q}$,

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1^1 & \cdots & \mathbf{q}_{m_q}^1 \\ \vdots & \ddots & \vdots \\ \mathbf{q}_1^n & \cdots & \mathbf{q}_{m_q}^n \end{bmatrix} \in \mathbb{R}^{3n \times m_q},$$

contains n frames and m_q joints. For instance, Fig. 3a shows a set of motion capture segments randomly selected from several kicking sequences.

To align the segments, we apply Procrustes analysis to remove the 3D rigid transformations. In order to build local models, we cluster all segments into k groups using spectral clustering [35]. The affinity between each pair of segments is computed as,

$$\kappa(\mathbf{Q}_i, \mathbf{Q}_j) = \exp\Big(-\frac{1}{\sigma^2}(\|\mathbf{Q}_i - \tau_{ij}(\mathbf{Q}_j)\|_F^2 + \|\mathbf{Q}_j - \tau_{ji}(\mathbf{Q}_i)\|_F^2)\Big),$$

where τ_{ij} denotes the similarity transformation found by Procrustes analysis when aligning $\mathbf{Q}_j$ towards $\mathbf{Q}_i$. The kernel bandwidth σ is set to be the average distance from the 50% closest neighbors for all $\mathbf{Q}_i$ and $\mathbf{Q}_j$ pairs. As shown in the experiments, this clustering step improves the generalization of the learned shape models. For instance, each of the 4 segment clusters shown in Fig. 3b corresponds to a different temporal stage of kicking a ball. Please refer Fig. 4 for examples of temporal clusters.

Given a set of l segments[2], $\{\mathbf{Q}_i\}_{i=1}^l$, belonging to each cluster, we learn a bilinear model [34] such that each segment $\mathbf{Q}_i$ can be reconstructed using a set of weights $\mathbf{W}_i \in \mathbb{R}^{k_t \times k_s}$ minimizing,

$$\min_{\mathbf{T},\mathbf{S},\{\mathbf{W}_i\}_i} \sum_{i=1}^{l} \|\mathcal{Q}(\mathbf{T}\mathbf{W}_i\mathbf{S}^T) - \mathbf{Q}_i\|_F^2, \tag{1}$$

[2] To simplify the notation, we do not explicitly specify the cluster membership of the motion capture segment ($\mathbf{Q}_i$) and the bilinear bases ($\mathbf{T}$ and $\mathbf{S}$).

where the columns of $\mathbf{T} \in \mathbb{R}^{n \times k_t}$ and $\mathbf{S} \in \mathbb{R}^{3m_q \times k_s}$ contain k_t trajectories and k_s shape bases respectively. In the experiment, we found $k_t = 10$ and $k_s = 15$ produced consistently good results. $\mathcal{Q}(\cdot)$ is a linear operator that reshapes any n-by-$3m_q$ matrix to a $3n$-by-m_q one, *i.e.*,

$$\mathcal{Q}\Big(\begin{bmatrix} \mathbf{q}_1^{1T} & \cdots & \mathbf{q}_{m_q}^{1T} \\ \vdots & \ddots & \vdots \\ \mathbf{q}_1^{nT} & \cdots & \mathbf{q}_{m_q}^{nT} \end{bmatrix}\Big) = \begin{bmatrix} \mathbf{q}_1^{1} & \cdots & \mathbf{q}_{m_q}^{1} \\ \vdots & \ddots & \vdots \\ \mathbf{q}_1^{n} & \cdots & \mathbf{q}_{m_q}^{n} \end{bmatrix}, \forall\ \mathbf{q}_j^i \in \mathbb{R}^3.$$

Unfortunately, optimizing Eq. 1 jointly over the bilinear bases $\mathbf{T}$, $\mathbf{S}$ and their weights $\{\mathbf{W}_i\}_i$ is a non-convex problem. To reduce the complexity and make the problem more trackable, we fix $\mathbf{T}$ to be the discrete cosine transform (DCT) bases (Top of Fig. 3c). Following [34], the shape bases $\mathbf{S}$ can then be computed in closed-form using the SVD as,

$$[\mathbf{T}\mathbf{T}^{\dagger}\mathcal{Q}^{-1}(\mathbf{Q}_1); \cdots; \mathbf{T}\mathbf{T}^{\dagger}\mathcal{Q}^{-1}(\mathbf{Q}_l)] = \mathbf{U}\mathbf{\Sigma}\mathbf{S}^T. \tag{2}$$

For example, the left part of Fig. 3c plots the first two shape bases $\mathbf{s}_i$ learned from the 3^{rd} cluster of segments shown in Fig. 3b, which mainly capture the deformation of the movements of the arms and legs.

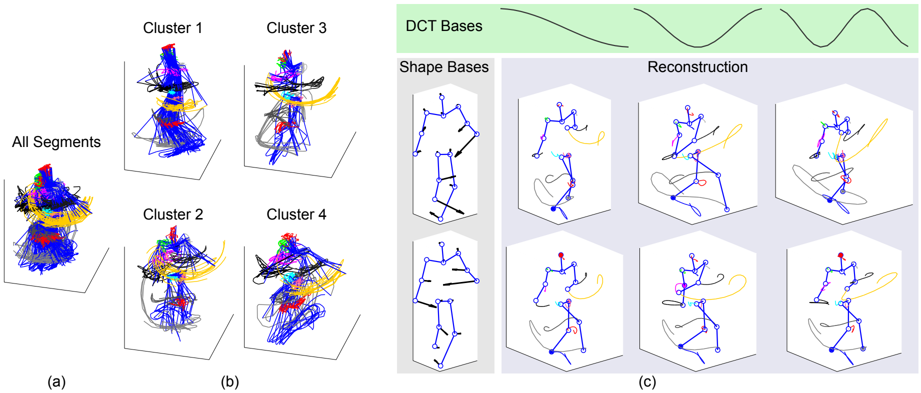

Fig. 3. Spatio-temporal bilinear model learned from the CMU motion capture dataset. (a) All the motion capture segments randomly selected from a set of kicking sequences. (b) Clustering motion capture segments into 4 temporal clusters. (c) The bilinear bases estimated from the 3^{rd} cluster. Left: top-2 shape bases ($\mathbf{s}_i$) where the shape deformation is visualized by black arrows. Top: top-3 DCT trajectory bases ($\mathbf{t}_j$). Bottom-right: bilinear reconstruction by combining each pair of shape and DCT bases ($\mathbf{t}_j\mathbf{s}_i^T$).

3.3 STM optimization

This section describes the objective function and the optimization strategy for the STM algorithm.

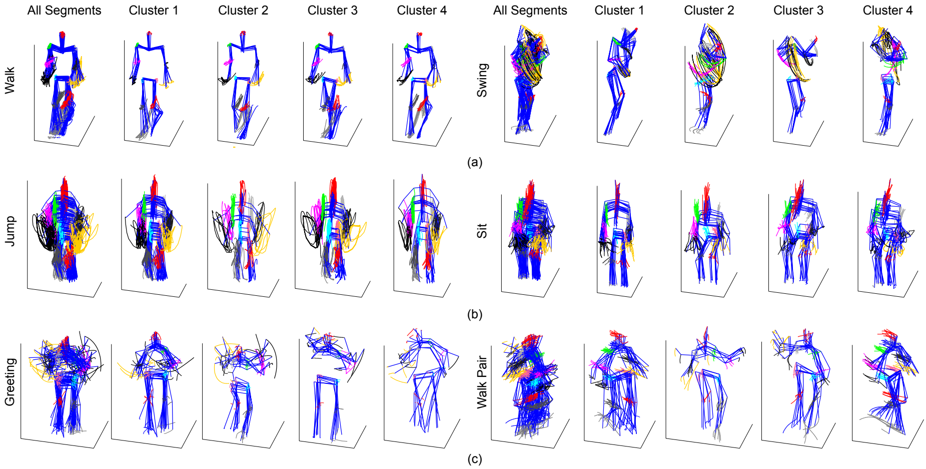

Fig. 4. Clustering motion capture segments into four clusters for different datasets. (a) CMU motion capture dataset [29]. (b) Berkeley MHAD dataset [10]. (c) Human3.6M dataset [9].

Objective function. Given the m_p trajectories $\mathbf{P} \in \mathbb{R}^{2n \times m_p}$ extracted from an n-length video segment, STM aims to select a subset of m_q trajectories that best fits the learned spatio-temporal 3D shape structure ($\mathbf{T}$ and $\mathbf{S}$) projected in 2D. More specifically, the problem of STM consists in finding a many-to-one correspondence matrix $\mathbf{X} \in \{0,1\}^{m_p \times m_q}$, weights of the bilinear 3D model $\mathbf{W} \in \mathbb{R}^{k_t \times k_s}$, and a set of 3D-2D weak perspective projections[3] $\mathbf{R} \in \mathbb{R}^{2n \times 3n}$, $\mathbf{b} \in \mathbb{R}^{2n}$, that minimize the following error

$$\begin{aligned} \min_{\mathbf{X},\mathbf{W},\mathbf{R},\mathbf{b}} \quad & \|\mathbf{R}\mathcal{Q}(\mathbf{T}\mathbf{W}\mathbf{S}^T) + \mathbf{b}\mathbf{1}^T - \mathbf{P}\mathbf{X}\|_1 + \lambda_a \operatorname{tr}(\mathbf{A}\mathbf{X}) + \lambda_s \|\mathbf{T}\mathbf{W}\mathbf{\Sigma}^{-1}\|_1, \\ \text{s. t.} \quad & \mathbf{X} \in \{0,1\}^{m_p \times m_q},\ \mathbf{X}^T\mathbf{1} = \mathbf{1},\ \mathbf{R}_i^T\mathbf{R}_i = \mathbf{I}_2\ \forall\ i = 1 \cdots n, \end{aligned} \tag{3}$$

where the first term in the objective measures the error between the selected trajectories $\mathbf{PX} \in \mathbb{R}^{2n \times m_q}$ and the bilinear reconstruction $\mathcal{Q}(\mathbf{T}\mathbf{W}\mathbf{S}^T)$ projected in 2D using $\mathbf{R}$ and $\mathbf{b}$. The error is computed using the l_1 norm instead of the Frobenious norm, because of its efficiency and robustness. Given the trajectory response $\mathbf{A} \in \mathbb{R}^{m_q \times m_p}$, the second term measures the appearance cost of the trajectories selected by $\mathbf{X}$ and weighted by λ_a. The third term weighted by λ_s penalizes large weights $\mathbf{TW} \in \mathbb{R}^{n \times k_s}$ of the shape bases, where the singular value $\mathbf{\Sigma} \in \mathbb{R}^{k_s \times k_s}$ computed in Eq. 2 is used to normalize the contribution of each basis. In our experiment, the regularization weights λ_a and λ_s are estimated using cross-validation.

Optimizing Eq. 3 is a challenging problem, in the following sections we describe an efficient coordinate-descent algorithm that alternates between solving

[3] $\mathbf{R} = [\searrow_i \theta_i \mathbf{R}_i] \in \mathbb{R}^{2n \times 3n}$ is a block-diagonal matrix, where each block contains the rotation $\mathbf{R}_i \in \mathbb{R}^{2 \times 3}$ and scaling θ_i for each frame. Similarly, $\mathbf{b} = [\mathbf{b}_1; \cdots; \mathbf{b}_n] \in \mathbb{R}^{2n}$ is a concatenation of the translation $\mathbf{b}_i \in \mathbb{R}^2$ for each frame.

$\mathbf{X}, \mathbf{W}$ and $\mathbf{R}, \mathbf{b}$ until convergence. The algorithm is initialized by computing $\mathbf{X}$ that minimizes the appearance cost $\operatorname{tr}(\mathbf{AX})$ in Eq. 3 and setting $\mathcal{Q}(\mathbf{TWS}^T)$ to be the mean of the motion capture segments.

Optimizing STM over X and W. Due to the combinatorial constraint on $\mathbf{X}$, optimizing Eq. 3 over $\mathbf{X}$ and $\mathbf{W}$ given $\mathbf{R}$ and $\mathbf{b}$ is a NP-hard mixed-integer problem. To approximate the problem, we relax the binary $\mathbf{X}$ to be a continuous one and reformulate the problem using the LP trick [36] as,

$$\begin{aligned}\min_{\mathbf{X},\mathbf{W},\mathbf{U},\mathbf{V},\mathbf{U}_s,\mathbf{V}_s} \quad & \mathbf{1}^T(\mathbf{U}+\mathbf{V})\mathbf{1} + \lambda_a \operatorname{tr}(\mathbf{AX}) + \lambda_s \mathbf{1}^T(\mathbf{U}_s+\mathbf{V}_s)\mathbf{1}, \qquad (4)\\ \text{s.t.} \quad & \mathbf{X} \in [0,1]^{m_p \times m_q}, \mathbf{X}^T\mathbf{1} = \mathbf{1},\\ & \mathbf{R}\mathcal{Q}(\mathbf{TWS}^T) + \mathbf{b}\mathbf{1}^T - \mathbf{PX} = \mathbf{U} - \mathbf{V}, \mathbf{U} \geq \mathbf{0}, \mathbf{V} \geq \mathbf{0},\\ & \mathbf{TW}\boldsymbol{\Sigma}^{-1} = \mathbf{U}_s - \mathbf{V}_s, \mathbf{U}_s \geq \mathbf{0}, \mathbf{V}_s \geq \mathbf{0},\end{aligned}$$

where $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{2n \times m_q}$ and $\mathbf{U}_s, \mathbf{V}_s \in \mathbb{R}^{n \times k_s}$ are four auxiliary variables used to formulate the l_1 problem as linear programming. The term $\mathbf{R}\mathcal{Q}(\mathbf{TWS}^T)$ is linear in $\mathbf{W}$ and we can conveniently re-write this expression using the following equality as:

$$\begin{aligned}&\operatorname{vec}\left(\mathbf{R}\mathcal{Q}(\mathbf{TWS}^T)\right) = (\mathbf{I}_{m_q} \otimes \mathbf{R}) \operatorname{vec}\left(\mathcal{Q}(\mathbf{TWS}^T)\right)\\ &= (\mathbf{I}_{m_q} \otimes \mathbf{R})\boldsymbol{\Pi}_{\mathcal{Q}} \operatorname{vec}(\mathbf{TWS}^T) \quad = \underbrace{(\mathbf{I}_{m_q} \otimes \mathbf{R})\boldsymbol{\Pi}_{\mathcal{Q}}(\mathbf{S} \otimes \mathbf{T})}_{\text{Constant}} \operatorname{vec}(\mathbf{W}),\end{aligned}$$

where $\boldsymbol{\Pi}_{\mathcal{Q}} \in \{0,1\}^{3nm_q \times 3nm_q}$ is a permutation matrix that re-orders the elements of a $3nm_q$-D vector as,

$$\boldsymbol{\Pi}_{\mathcal{Q}} \operatorname{vec}\left(\begin{bmatrix} \mathbf{q}_1^1 & \cdots & \mathbf{q}_{m_q}^1 \\ \vdots & \ddots & \vdots \\ \mathbf{q}_1^n & \cdots & \mathbf{q}_{m_q}^n \end{bmatrix}\right) = \operatorname{vec}\left(\begin{bmatrix} \mathbf{q}_1^{1T} & \cdots & \mathbf{q}_{m_q}^{1T} \\ \vdots & \ddots & \vdots \\ \mathbf{q}_1^{nT} & \cdots & \mathbf{q}_{m_q}^{nT} \end{bmatrix}\right), \forall\, \mathbf{q}_j^i \in \mathbb{R}^3. \qquad (5)$$

After solving the linear program, we gradually discretize $\mathbf{X}$ by taking successive refinements based on trust-region shrinking [36].

Optimizing STM over R and b. If $\mathbf{X}$ and $\mathbf{W}$ are fixed, optimizing Eq. 3 with respect to $\mathbf{R}$ and $\mathbf{b}$ becomes an l_1 Procrustes problem [37],

$$\min_{\mathbf{R},\mathbf{b}} \quad \|\mathbf{RQ} + \mathbf{b}\mathbf{1}^T - \mathbf{PX}\|_1, \quad \text{s.t. } \mathbf{R}_i^T\mathbf{R}_i = \mathbf{I}_2 \ \forall\, i = 1 \cdots n, \qquad (6)$$

where $\mathbf{Q} = \mathcal{Q}(\mathbf{TWS}^T)$. Inspired by the recent advances in compressed sensing, we approximate Eq. 6 using the augmented Lagrange multipliers method [38] that minimizes the following augmented Lagrange function:

$$\begin{aligned}\min_{\mathbf{L},\mathbf{E},\mu,\mathbf{R},\mathbf{b}} \ & \|\mathbf{E} - \mathbf{PX}\|_1 + \operatorname{tr}\left(\mathbf{L}^T(\mathbf{RQ} + \mathbf{b}\mathbf{1}^T - \mathbf{E})\right) + \frac{\mu}{2}\|\mathbf{RQ} + \mathbf{b}\mathbf{1}^T - \mathbf{E}\|_F^2, \qquad (7)\\ & \text{s.t. } \mathbf{R}_i^T\mathbf{R}_i = \mathbf{I}_2 \ \forall\, i = 1 \cdots n,\end{aligned}$$

where $\mathbf{L}$ is the Lagrange multiplier, $\mathbf{E}$ is an auxiliary variable, and μ is the penalty parameter. Eq. 7 can be efficiently approximated in a coordinate-descent manner. First, optimizing Eq. 7 with respect to $\mathbf{R}$ and $\mathbf{b}$ is a standard orthogonal Procrustes problem,

$$\min_{\mathbf{R},\mathbf{b}} \quad \|\mathbf{R}\mathbf{Q} + \mathbf{b}\mathbf{1}^T - (\mathbf{E} - \frac{\mathbf{L}}{\mu})\|_F^2, \quad \text{s.t. } \mathbf{R}_i^T\mathbf{R}_i = \mathbf{I}_2 \; \forall \; i = 1 \cdots n,$$

which has a close-form solution using the SVD. Second, optimizing Eq. 7 with respect to $\mathbf{E}$ can be efficiently found using absolute value shrinkage [38] as,

$$\mathbf{E} := \mathbf{P}\mathbf{X} - \mathcal{S}_{\frac{1}{\mu}}(\mathbf{P}\mathbf{X} - \mathbf{R}\mathbf{Q} - \mathbf{b}\mathbf{1}^T - \frac{\mathbf{L}}{\mu}),$$

where $\mathcal{S}_\sigma(p) = \max(|p| - \sigma, 0)\,\mathrm{sign}(p)$ is a soft-thresholding operator [38]. Third, we gradually update $\mathbf{L} \leftarrow \mathbf{L} + \mu(\mathbf{R}\mathbf{Q} + \mathbf{b}\mathbf{1}^T - \mathbf{E})$ and $\mu \leftarrow \rho\mu$, where we set the incremental ratio to $\rho = 1.05$ in all our experiments.

3.4 Fusion

Given a video containing an arbitrary number of frames, we solved STM independently for each segment of n frames ($n = 15$ in our experiments). Recall that we learned k bilinear models ($\mathbf{T}$ and $\mathbf{S}$) from different clusters of motion capture segments (*e.g.*, Fig. 3b) in the training step. To find the best model for each segment, we optimize Eq. 3 using each model and select the one with the smallest error. After solving STM for each segment, the final joint position $\bar{\mathbf{P}}_i \in \mathbb{R}^{2\times m_q}$ at frame i is the average coordinates of the selected trajectories $\{\mathbf{P}_c\mathbf{X}_c\}_c$ from all the l_c segments overlapped at i, *i.e.*, $\bar{\mathbf{P}}_i = \frac{1}{l_c}\sum_c \mathbf{P}_c^{i_c}\mathbf{X}_c$, where $\mathbf{P}_c^{i_c} \in \mathbb{R}^{2\times m_p}$ encodes the trajectory coordinates at the i_c^{th} frame within the c^{th} segment and i_c the local index of the i^{th} frame in the original video.

4 Experiments

This section compares STM against several state-of-the-art algorithms for body part detection in synthetic experiments on the CMU motion capture dataset [29], and real experiments on the MHAD [10] and the Human3.6M [9] datasets.

For each dataset, the 3D motion capture model was trained from its associated motion capture sequences. The 3D motion capture training data is person-independent, and it does not contains samples of the testing subject. Notice that the annotation scheme is different across datasets (Fig. 2a). We investigated four different types of 3D models for STM: (1) Generic models: **STM-G1** and **STM-G4** were trained using all sequences of different actions with $k = 1$ and $k = 4$ clusters respectively. (2) Action-specific models: **STM-A1** and **STM-A4** were trained independently for each action from each dataset. In testing, we assumed we know what action the subject was performing. As before, **STM-A1** and **STM-A4** were trained with $k = 1$ and $k = 4$ clusters respectively.

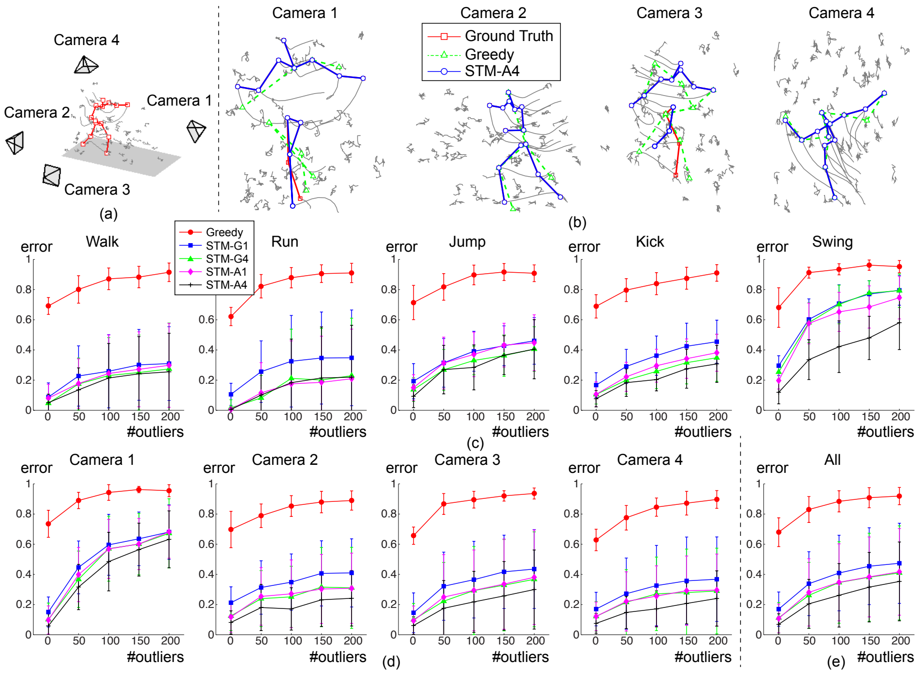

Fig. 5. Comparison of human pose estimation on the CMU motion capture dataset. (a) Original motion capture key-frames in 3D with 50 outliers that were synthetically generated. (b) Results of the greedy approach and our method on four 2D projections. (c) Mean error and std. for each method and action as a function of the number of outliers. (d) Mean error and std. for each camera view. (e) Mean error and std. for all actions and cameras.

4.1 CMU motion capture dataset

The first experiment validated our approach on the CMU motion capture dataset [29], from which we selected 5 actions including walking, running, jumping, kicking, golf swing. For each action, we picked 8 sequences performed by different subjects. For each sequence, we synthetically generated $0 \sim 200$ random trajectories as outliers in 3D. Then we projected each sequence (with outliers included) onto 4 different 2D views. See Fig. 5a for examples of the 3D sequences as well as the camera positions. To reproduce the response of a body part detector at each frame, we synthetically generate a constant-value response region centered at the ground-truth location with the radius being the maximum limb length over the sequence. The response value of the j^{th} feature trajectory for the c^{th} body part at i^{th} frame is considered to be $a_{cj}^{i} = -1$ if it falls in the region or 0 otherwise. Our goal is to detect the original trajectories and recover the body structure.

We quantitatively evaluated our method with a leave-one-out scheme, *i.e.*, each testing sequence was taken out for testing, and the remaining data was used for training the bilinear model. For each sequence, we computed the error of each method as the percentage of incorrect detections of the feature points compared with the ground-truth position averaged over frames. To the best of our knowledge, there is no previous work on STM in computer vision. Therefore, we implemented a greedy baseline that selects the optimal feature points with the lowest response cost without geometrical constraints.

Fig. 5b shows some key-frames for the greedy approach, our method and the ground truth using the STM-A4 for detecting the kicking actions across four views. As can be observed, STM is able to select the trajectories more precisely and it is more robust than the greedy approach. Fig. 5c-d quantitatively compare our methods with the greedy approach on each action and viewpoint respectively. Our method consistently outperforms the greedy approach for detection and tracking in presence of outliers. In addition, the STM-A1 model obtains lower error rates than STM-G1 because STM-A1 is an action-specific model, unlike STM-G1 which is a generic one. By increasing the number of clusters from one to four, the performance of STM-G4 and STM-A4 clearly improves from STM-G1 and STM-A1 respectively. This not surprising because the bilinear models trained on a group of similar segments can be represented more compactly (fewer number of parameters) and generalize better in testing.

4.2 Berkeley multi-modal human action (MHAD) dataset

In the second experiment, we tested the ability of STM to detect humans on the Berkeley multi-modal human action database (MHAD) [10]. The MHAD database contains 11 actions performed by 12 subjects. For each sequence, we took the videos captured by 2 different cameras as shown in Fig. 6a. To extract the trajectories from each video, we used [25] in sliding-window manner to extract dense trajectories from each 15 frames segment. The response for each trajectory was computed using the SVM detector score [5]. The bilinear models were trained from the motion capture data associated with this dataset.

To quantitatively evaluate the performance, we compared our method with two baselines: the state-of-the-art image-based pose estimation method proposed by Yang and Ramanan [5], and the recent video-based method designed by Burgos *et al.* [15] that merges multiple independent pose estimates across frames. We evaluated all methods with a leave-one-out scheme. The error for each method is computed as the pixel distance between the estimated and ground-truth part locations. Notice that a portion of the error is due to the inconsistency in labeling protocol between the PARSE model [5] and the MHAD dataset.

Fig. 6b-d compare the error to localize body parts of our method againts [5] and [15]. Our method largely improves the image-based baseline [5] for all actions and viewpoints. Compared to the video-based method [15], STM achieves lower errors for most actions except for "jump jacking", "bending", "one-hand waving" and "two-hand waving", where the fast movement of the body joints cause much larger error in tracking feature trajectories over time. Among the four STM models, STM-A4 performs the best because the clustering step improves the generalization of the bilinear model. As shown in Fig. 6d, the hands are the most difficult to accurately detect because of their fast movements and frequent occlusions.

Fig. 6e-g investigate the three main parameters of our system, segment length (n), number of bases (k_s and k_t) and the regularization weights (λ_a and λ_s). According to Fig. 6e, a smaller segment length is beneficial for "jump jacking" because the performance of the tracker [25] is less stable for fast-speed action. In contrast, using a larger window improves the temporal consistency in actions such as "throwing" and "standing up". Fig. 6f shows the detection error of STM using different number of shape (k_s) and trajectories (k_t) bases for the first subject. Overall, we found the performance of STM is not very sensitive to small change in the number of shape bases because the contribution of each shape basis in STM (Eq. 3) is normalized by their energies ($\mathbf{\Sigma}$). In addition, using a small number (*e.g.*, 5) of trajectory bases can lower the performance of STM. This result demonstrates the effectiveness of using dynamic models over the static ones (*e.g.*, a PCA-based model can be considered as a special case of the bilinear model when $k_t = 1$). Fig. 6g plots the cross-validation error for the first subject, from which we pick the optimal λ_a and λ_s.

Our system was implemented in Matlab on a PC with 2GHz Intel CPU and 8GB memory. The codes of [5, 15] were downloaded from authors' webpages. The linear programming in Eq. 4 was optimized using the Mosek LP solver [39]. Fig. 6f analyzes the computational cost (in seconds) for tracking the human pose in a sequence containing 126 frames. The most computationally intensive part of the method is calculating the response for each joint and each frame using [5]. Despite a large number of candidate trajectories ($m_p \approx 700$) per segment, STM can be computing in about 8 minutes.

4.3 Human3.6M dataset

In the last experiment, we selected 11 actions performed by 5 subjects from the Human3.6M dataset [9]. Compared to the Berkeley MHAD dataset, the motions in Human3.6M were performed by professional actors, that wear regular clothing to maintain as much realism as possible. See Fig. 7a for example frames.

As in the previous experiment, our methods were compared with two baselines [5, 15] in a leave-one-out scheme. The bilinear models were trained from the motion capture data associated with this dataset. Fig. 6b-c show the performance of each method on localizing body part for each action and viewpoint respectively. Due to the larger appearance variation and more complex motion performance, the overall error of each method is larger than the one achieved on the previous Berkeley MHAD dataset. However, STM still outperforms both the baselines [5, 15] for most actions and viewpoints. If the action label is known a priori, training action-specific models (STM-A1 and STM-A4) achieves better performance than the ones trained on all actions (STM-G1 and STM-G4).

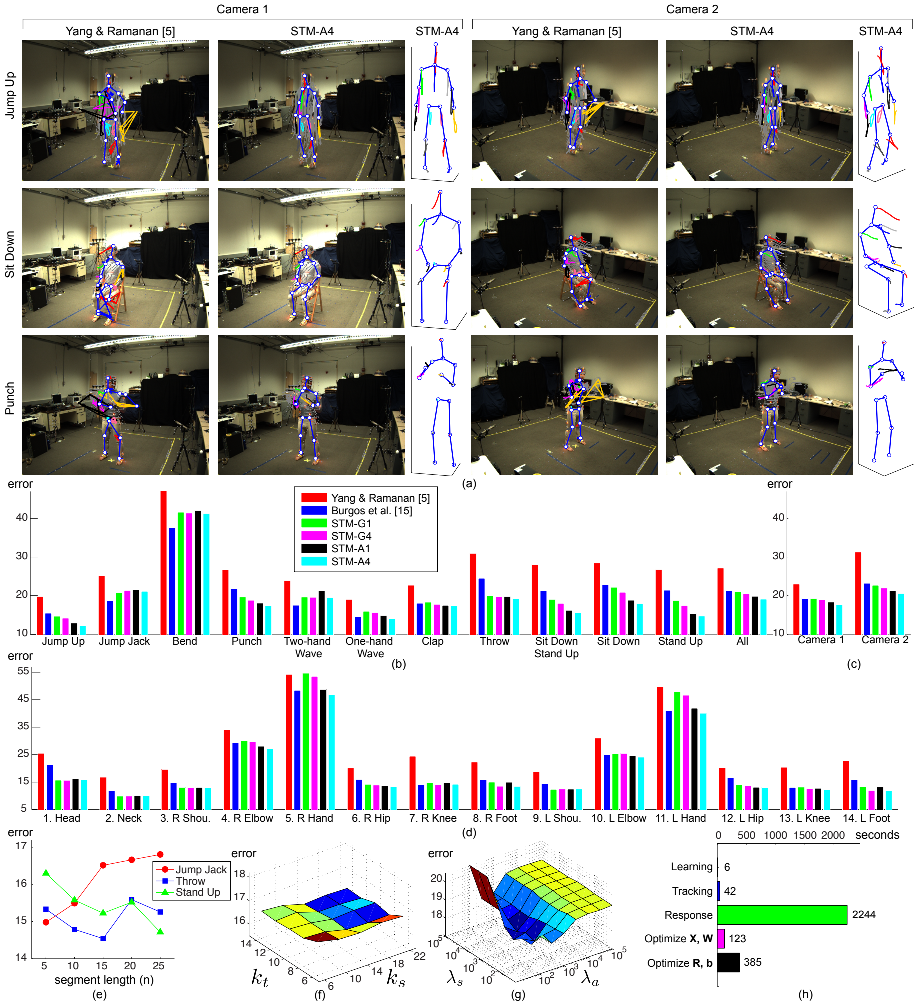

Fig. 6. Comparison of human pose estimation on the Berkeley MHAD dataset. (a) Result of [5] and our method on three actions of two views, where the 3D reconstruction estimated by our method is plotted on the right. (b) Errors for each action. (c) Errors for each camera view. (d) Errors of each joint. (e) Errors with respect to the segment length (n). (f) Errors with respect to the bases number (k_s and k_t). (g) Errors with respect to the regularization weights (λ_a and λ_s). (h) Time cost of each step.

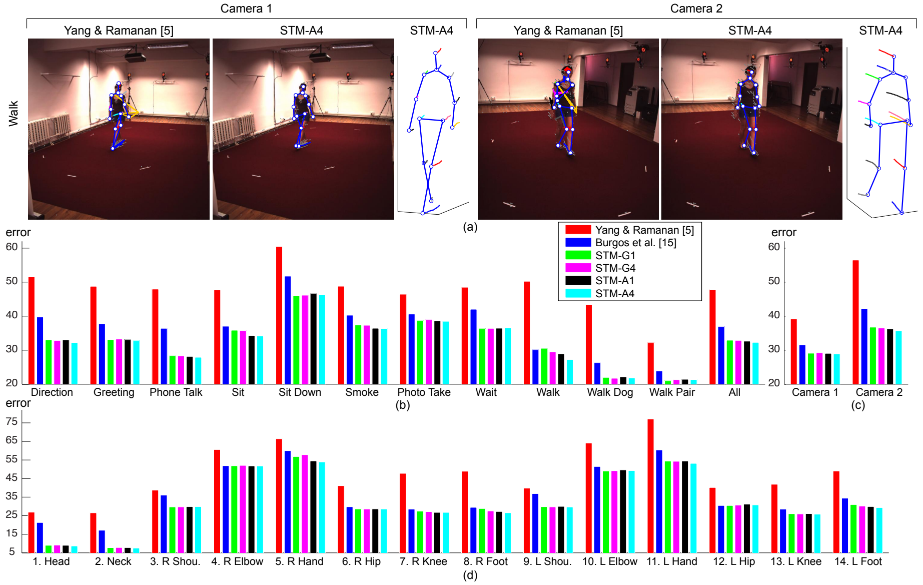

Fig. 7. Comparison of human pose estimation on the Human 3.6M dataset. (a) Result of [5] and our method on three actions of two views, where the 3D reconstruction estimated by our method is plotted on the right. (b) Errors for each action. (c) Errors for each camera view. (d) Errors of each joint.

5 Conclusion

This paper presents STM, a robust method for detection and tracking human poses in videos by matching video trajectories to a 3D motion capture model. STM matches trajectories to a 3D model, and hence it provides intrinsic view-invariance. The main novelty of the work resides in computing the correspondence between video and motion capture data. Although it might seem computationally expensive and difficult to optimize at first, using an l_1-formulation to solve for correspondence results in an algorithm that is efficient and robust to outliers, missing data and mismatches. We showed how STM outperforms state-of-the-art approaches to object detection based on deformable parts models in the (MHAD) [10] and the Human3.6M dataset [9].

A major limitation of our current approach is the high computational cost for calculating the joint response, which is computed independently for each frame. In future work, we plan to incorporate richer temporal features [25] to improve the speed and accuracy of the trajectory response. Also, we are solving STM independently for each segment (sub-sequence), which might result in some discontinuity in the estimation of the pose; a straight-forward improvement could be made by imposing consistency between overlapping segments.

References

1. Shotton, J., Girshick, R.B., Fitzgibbon, A.W., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., Blake, A.: Efficient human pose estimation from single depth images. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12) (2013) 2821–2840
2. Boiman, O., Irani, M.: Detecting irregularities in images and in video. Int. J. Comput. Vis. **74**(1) (2007) 17–31
3. Wei, X.K., Chai, J.: VideoMocap: modeling physically realistic human motion from monocular video sequences. ACM Trans. Graph. **29**(4) (2010)
4. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9) (2010) 1627–1645
5. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12) (2013) 2878–2890
6. Andriluka, M., Roth, S., Schiele, B.: Discriminative appearance models for pictorial structures. Int. J. Comput. Vis. **99**(3) (2012) 259–280
7. Ionescu, C., Li, F., Sminchisescu, C.: Latent structured models for human pose estimation. In: ICCV. (2011)
8. Eichner, M., Jesús, M., Zisserman, A., Ferrari, V.: 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. Int. J. Comput. Vis. **99**(2) (2012) 190–214
9. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6M: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Trans. Pattern Anal. Mach. Intell. (2014)
10. Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., Bajcsy, R.: Berkeley MHAD: A comprehensive multimodal human action database. In: IEEE Workshop on Applications on Computer Vision (WACV). (2013) 53–60
11. Poppe, R.: Vision-based human motion analysis: An overview. Comput. Vis. Image Underst. **108**(1-2) (2007) 4–18
12. Sapp, B., Toshev, A., Taskar, B.: Cascaded models for articulated pose estimation. In: ECCV. (2010)
13. Andriluka, M., Roth, S., Schiele, B.: Monocular 3d pose estimation and tracking by detection. In: CVPR. (2010)
14. Sapp, B., Weiss, D., Taskar, B.: Parsing human motion with stretchable models. In: CVPR. (2011)
15. Burgos, X., Hall, D., Perona, P., Dollár, P.: Merging pose estimates across space and time. In: BMVC. (2013)
16. Tian, Y., Sukthankar, R., Shah, M.: Spatiotemporal deformable part models for action detection. In: CVPR. (2013)
17. Zuffi, S., Romero, J., Schmid, C., Black, M.J.: Estimating human pose with flowing puppets. In: ICCV. (2013)
18. Agarwal, A., Triggs, B.: Recovering 3d human pose from monocular images. IEEE Trans. Pattern Anal. Mach. Intell. **28**(1) (2006) 44–58
19. Elgammal, A.M., Lee, C.S.: Inferring 3d body pose from silhouettes using activity manifold learning. In: CVPR. (2004)
20. Urtasun, R., Fleet, D.J., Fua, P.: 3d people tracking with Gaussian process dynamical models. In: CVPR. (2006)
21. Sigal, L., Black, M.J.: Predicting 3d people from 2d pictures. In: AMDO. (2006)

22. Simo-Serra, E., Ramisa, A., Alenyà, G., Torras, C., Moreno-Noguer, F.: Single image 3d human pose estimation from noisy observations. In: CVPR. (2012)
23. Yao, A., Gall, J., Gool, L.J.V.: Coupled action recognition and pose estimation from multiple views. Int. J. Comput. Vis. **100**(1) (2012) 16–37
24. Yu, T.H., Kim, T.K., Cipolla, R.: Unconstrained monocular 3d human pose estimation by action detection and cross-modality regression forest. In: CVPR. (2013)
25. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. Int. J. Comput. Vis. **103**(1) (2013) 60–79
26. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: SCIA. (2003)
27. Messing, R., Pal, C.J., Kautz, H.A.: Activity recognition using the velocity histories of tracked keypoints. In: ICCV. (2009)
28. Matikainen, P., Hebert, M., Sukthankar, R.: Trajectons: Action recognition through the motion analysis of tracked features. In: ICCVW. (2009)
29. Carnegie Mellon University Motion Capture Database. http://mocap.cs.cmu.edu
30. Park, D., Ramanan, D.: N-best maximal decoders for part models. In: ICCV. (2011)
31. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical geometry of non-rigid shapes. Springer (2008)
32. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3d shape from image streams. In: CVPR. (2000)
33. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Trajectory space: A dual representation for nonrigid structure from motion. IEEE Trans. Pattern Anal. Mach. Intell. **33**(7) (2011) 1442–1456
34. Akhter, I., Simon, T., Khan, S., Matthews, I., Sheikh, Y.: Bilinear spatiotemporal basis models. ACM Trans. Graph. **31**(2) (2012) 17
35. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: NIPS. (2001) 849–856
36. Jiang, H., Drew, M.S., Li, Z.N.: Matching by linear programming and successive convexification. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6) (2007) 959–975
37. Trendafilov, N.: On the l_1 Procrustes problem. Future Generation Computer Systems **19**(7) (2004) 1177–1186
38. Lin, Z., Chen, M., Ma, Y.: The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint arXiv:1009.5055 (2010)
39. Mosek. http://www.mosek.com/