Spatio-temporal Matching for Human Pose Estimation in Video

Feng Zhou and Fernando De la Torre

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 CMU motion capture, Human3.6M, Berkeley MHAD and CMU MAD databases illustrate the benefits of our method over state-of-the-art approaches.

Index Terms—Human pose estimation, Dense trajectories, Spatio-temporal bilinear model, Trajectory matching

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 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 Berkeley MHAD database [9], the Human3.6M database [10] and Multi-modal action dataset [11], 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

This section reviews related works in human detection in video and 3D human pose estimation.

2.1 Human detection in video

A review of the literature on people tracking is well beyond the scope of this paper. We focus our attention here on the work most similar in spirit to ours. Many early approaches [12]–[18] were based on simple appearance models (*e.g.*, silhouettes) and performed tracking using stochastic search with kinematic constraints. However, silhouette extraction becomes unreliable because of complex backgrounds, occlusions, and moving cameras. Moreover, stochastic search in these high-dimensional spaces is notoriously difficult.

Facilitated by the advances in human detection methods [4]–[7], [19], tracking by detection has been a focus of recent work. For instance, Andriluka *et al.* [20], [21] combined the initial estimate of the human pose across frames in a tracking-by-detection framework. Sapp *et*

F. Zhou and F. De la Torre are with Robotics Institute, Carnegie Mellon University.

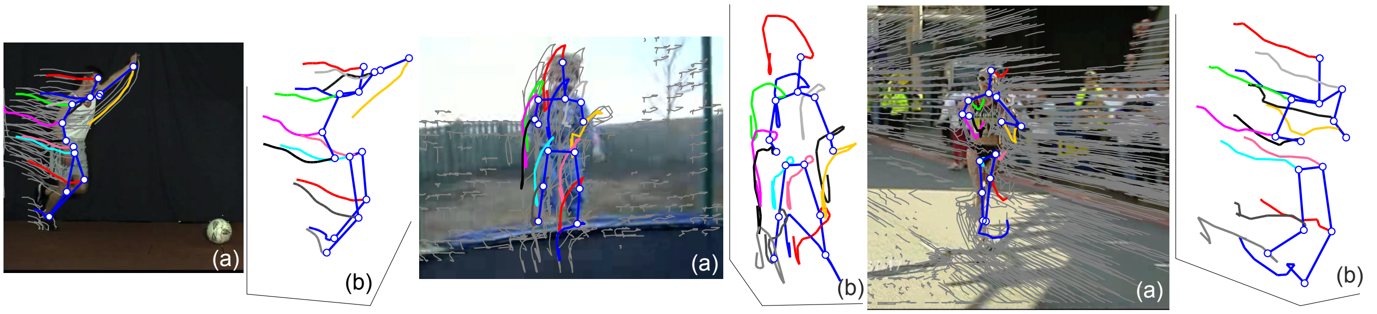


Fig. 1. Detection and tracking of humans in three videos using spatio-temporal matching (STM). STM extracts trajectories in video (gray lines) and selects a subset of trajectories (a) that match with the 3D motion capture model (b) learned from the CMU motion capture data set. Better viewed in color.

al. [22] coupled locations of body joints within and across frames from an ensemble of tractable sub-models. Wu and Nevatia [23] propose an approach for detecting and tracking partially occluded people using an assembly of body parts. Such tracking-by-detection approaches are attractive because they can avoid drift and recover from errors. The most similar work to ours are the recent fusion method by stitching together N-best hypotheses from frames of a video. Burgos *et al.* [24] merged multiple independent pose estimates across space and time using a non-maximum suppression. Park and Ramanan [25] generated multiple diverse high-scoring pose proposals from a tree-structured model and used a chain CRF to track the pose through the sequence. Inspired by the recent success on using convolutional neural network (CNN) [26] for the task of human body pose detection, Jain *et al.*. [27] proposed MoDeep for articulated human pose estimation in videos using a CNN architecture, which incorporates both color and motion features. Compared to these methods, our work enforces temporal consistency by matching video trajectories to a spatio-temporal 3D model, and provide robustness to view-point changes.

2.2 3D human pose estimation

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 [28]) to 3D pose with different priors [29], [30]. 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], [5], [19]. For instance, Sigal and Black [31] learned a mixture of experts model to infer 3D poses conditioned on 2D poses. Simo-Serra *et al.* [32] 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 [33], Yu *et al.* [34] 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 System overview

This section describes an overview of our proposed spatial-temporal matching (STM) method. The overview of the method is illustrated in Fig. 2. The STM algorithm has three main components. **Section. 4**: Given an input video, STM extracts 2D feature trajectories and evaluates the pseudo-likelihood of each pixel belonging to different body parts. **Section. 5**: During training, STM learns a bilinear spatio-temporal 3D model from motion capture data that will be used to constraint possible video trajectories. **Section. 6**: During testing, STM finds a subset of trajectories that correspond to 3D joints in the spatio-temporal model, and compute the extrinsic camera parameters.

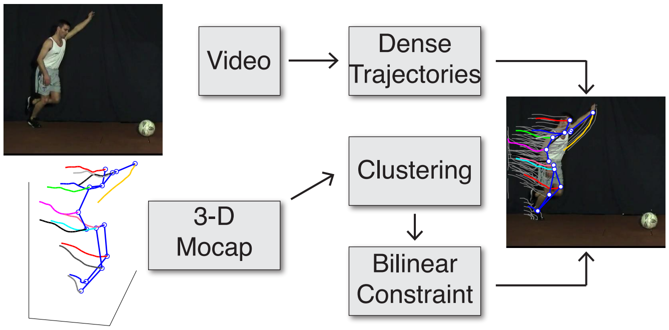


Fig. 2. Overview of the STM method. Given a video, STM extracts video features and selects a subset of video features that match a 3D motion capture model.

4 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 [35] to extract trajectories by densely sampling feature points in the first frame and track them using a dense optical flow algorithm [36]. 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. 3b illustrates a video segment with densely extracted feature trajectories.

Compared to the sparser KLT-based trackers [37], [38], 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. Compared to the various spatio-temporal descriptors (*e.g.*, STIP [39], Cuboids [40]), trajectories capture more local motion information of the video. see [41] for a review.

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. 3a) that are common across several datasets including the PARSE human body model [5], CMU [42], Berkeley MHAD [9], Human3.6M [10] motion capture datasets and CMU MAD Kinect dataset [11].

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 [25]. Fig. 3c 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.

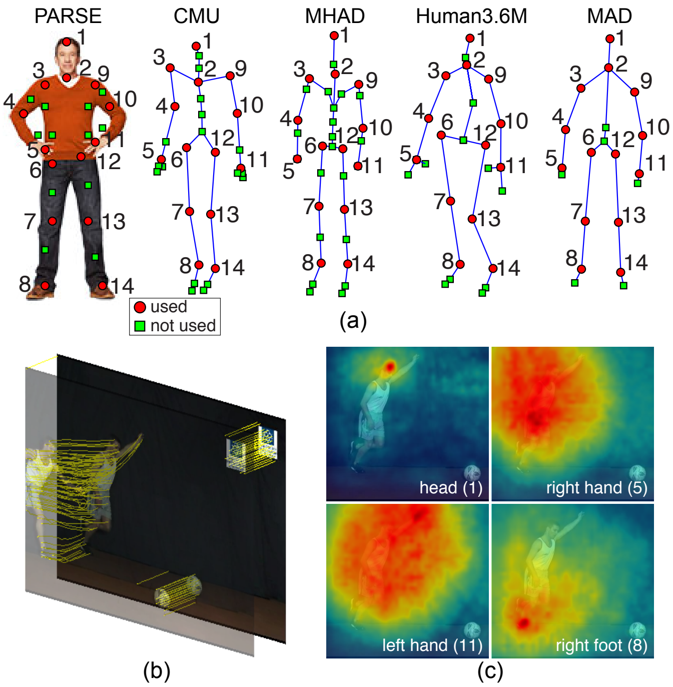


Fig. 3. Example of feature trajectories and their responses. (a) Geometrical configuration of 14 body joints shared across 3D datasets. (b) Dense trajectories extracted from a video segment. (c) Feature response maps for 4 joints (see bottom-right corner).

5 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 [43]. Common models include learning linear basis vectors independently for each frame [44] or discrete cosine transform bases independently for each joint trajectory [45]. 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 [46].

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. 4a shows a set of motion capture segments randomly selected from several kicking sequences.

To align the segments, we apply Procrustes analysis [47] to remove the 3D rigid transformations. In order to build local models, we cluster all segments

[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. $\operatorname{vec}(\mathbf{X})$ denotes the vectorization of matrix $\mathbf{X}$. $\mathbf{X} \circ \mathbf{Y}$ and $\mathbf{X} \otimes \mathbf{Y}$ are the Hadamard and Kronecker products of matrices.

into k groups using spectral clustering [48]. The affinity between each pair of segments is computed as,

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

where $\tau_{ij}(\cdot)$ 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. 4b corresponds to a different temporal stage of kicking a ball. Please refer Fig. 5 for more 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 [46] 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,

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

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[3] that reshapes any n-by-$3m_q$ matrix to a $3n$-by-m_q one, *i.e.,*

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

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. 4c). Following [46], 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. 4c plots the first two shape bases $\mathbf{s}_i$ learned from the 3^{rd} cluster of segments shown in Fig. 4b, which mainly capture the deformation of the movements of the arms and legs.

6 Spatio-temporal Matching (STM)

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

6.1 STM's Objective

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 three variables: (1) a binary correspondence matrix $\mathbf{X} \in \{0,1\}^{m_p \times m_q}$ under the many-to-one constraint $\mathbf{X}^T\mathbf{1} = \mathbf{1}$; (2) the weights $\mathbf{W} \in \mathbb{R}^{k_t \times k_s}$ of the bilinear 3D model; and (3) a set of 3D-2D weak perspective projections[4] $\mathbf{R} \in \mathbb{R}^{2n \times 3n}$, $\mathbf{b} \in \mathbb{R}^{2n}$, where the rotation needs to satisfy the orthogonal constraints,

$$\Psi = \Bigg\{ \mathbf{R} = \begin{bmatrix} \theta_1\mathbf{R}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \theta_n\mathbf{R}_n \end{bmatrix} \ \Bigg|\ \mathbf{R}_i^T\mathbf{R}_i = \mathbf{I}_2\ \forall\ i \Bigg\}. \tag{3}$$

In a nutshell, STM aims to solve the following problem

$$\begin{aligned} \min_{\mathbf{X},\mathbf{W},\mathbf{R},\mathbf{b}} \quad & \|\mathbf{R}\mathcal{Q}(\mathbf{T}\mathbf{W}\mathbf{S}^T) + \mathbf{b}\mathbf{1}^T - \mathbf{P}\mathbf{X}\|_1 + \lambda_a \operatorname{tr}(\mathbf{A}\mathbf{X}) \\ & + \lambda_s\|\mathbf{T}\mathbf{W}\mathbf{\Sigma}^{-1}\|_1 + \lambda_o\|\mathbf{G}\mathbf{P}\mathbf{X} - \mathbf{G}'\mathbf{P}'\mathbf{X}'\|_1, \qquad (4) \\ \text{s.t.}\ \ & \mathbf{X} \in \{0,1\}^{m_p \times m_q},\ \mathbf{X}^T\mathbf{1} = \mathbf{1}, \mathbf{R} \in \Psi, \end{aligned}$$

where the objective is composed by four terms. (1) The first term measures the error between the selected trajectories $\mathbf{P}\mathbf{X} \in \mathbb{R}^{2n \times m_q}$ 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. (2) 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. (3) The third term weighted by λ_s penalizes large weights $\mathbf{T}\mathbf{W} \in \mathbb{R}^{n \times k_s}$ 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. (4) To impose temporal continuity on the solution, the fourth term weighted by λ_o penalizes the l_1 distance between the reconstruction for the current segment $\mathbf{P}\mathbf{X}$ and the previous one $\mathbf{P}'\mathbf{X}'$, where $\mathbf{G} \in \{0,1\}^{2n_o \times 2n}$ and $\mathbf{G}' \in \{0,1\}^{2n_o \times 2n'}$ are two selection matrices that select the overlapped n_o frames between $\mathbf{P}\mathbf{X}$ and $\mathbf{P}'\mathbf{X}'$ respectively. In our experiment, the regularization weights λ_a, λ_s and λ_o are estimated using cross-validation.

Optimizing Eq. 4 is a challenging problem, in the following sections we describe an efficient coordinate-descent algorithm that alternates between solving $\mathbf{X}, \mathbf{W}$ and $\mathbf{R}, \mathbf{b}$ until convergence. The algorithm is initialized by computing $\mathbf{X}$ that minimizes the appearance cost $\operatorname{tr}(\mathbf{A}\mathbf{X})$ in Eq. 4 and setting $\mathcal{Q}(\mathbf{T}\mathbf{W}\mathbf{S}^T)$ to be the mean of the motion capture segments.

6.2 Optimizing STM over X and W

Due to the combinatorial constraint on $\mathbf{X}$, optimizing Eq. 4 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

[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 $\mathbf{S}$).

[3] $\mathcal{Q}(\mathbf{Q})$ can be written in matrix form as $\big((\mathbf{1}_{n \times m_q} \otimes \mathbf{I}_3) \circ (\mathbf{Q} \otimes \mathbf{1}_3)\big)(\mathbf{I}_{m_q} \otimes \mathbf{1}_3)$ for any $\mathbf{Q} \in \mathbb{R}^{n \times 3m_q}$.

[4] $\mathbf{R} = [\searrow_i \theta_i\mathbf{R}_i] \in \mathbb{R}^{2n \times 3n}$ 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.

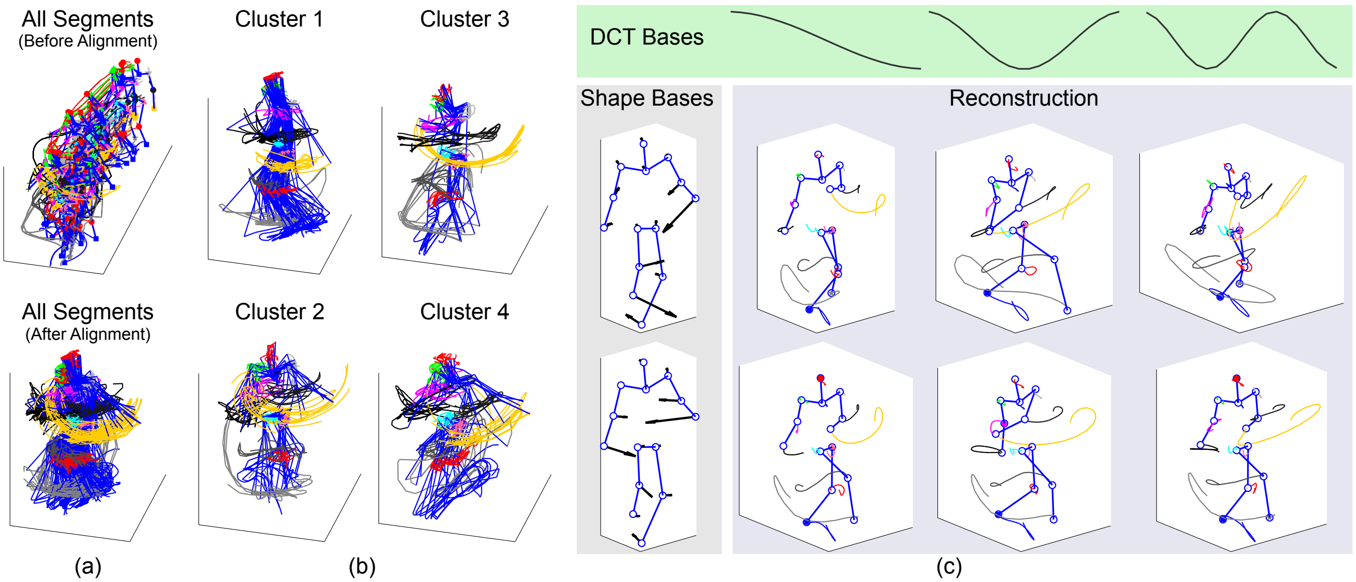


Fig. 4. Spatio-temporal bilinear model learned from the CMU motion capture dataset. (a) Top: All the motion capture segments randomly selected from a set of kicking sequences. Bottom: The segments are spatially aligned via Procrustes alignment. (b) Clustering motion capture segments into 4 temporal clusters. (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 $(\mathbf{t}_j\mathbf{s}_i^T)$.

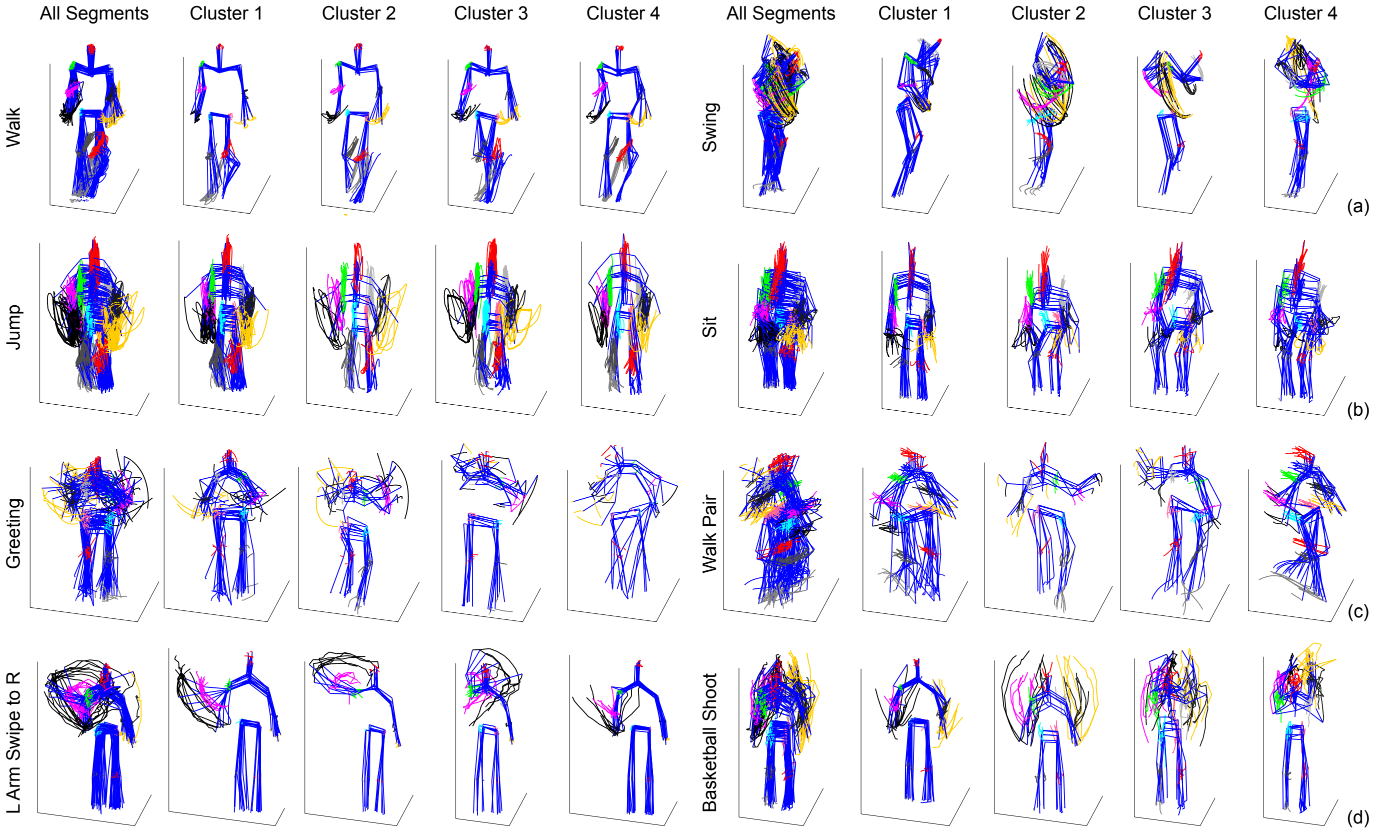


Fig. 5. Clustering motion capture segments into four clusters for different datasets. (a) CMU motion capture dataset [42]. (b) Berkeley MHAD dataset [9]. (c) Human3.6M dataset [10]. (d) CMU MAD dataset [11].

the binary $\mathbf{X}$ to be a continuous one and reformulate the problem using the LP trick[5] [49] as,

$$\begin{aligned}\min_{\substack{\mathbf{X},\mathbf{W},\mathbf{U},\mathbf{V}\\ \mathbf{U}_s,\mathbf{V}_s,\mathbf{U}_o,\mathbf{V}_o}} \quad & \mathbf{1}^T(\mathbf{U}+\mathbf{V})\mathbf{1}+\lambda_a \operatorname{tr}(\mathbf{AX})\\ & +\lambda_s\mathbf{1}^T(\mathbf{U}_s+\mathbf{V}_s)\mathbf{1}+\lambda_o\mathbf{1}^T(\mathbf{U}_o+\mathbf{V}_o)\mathbf{1}, \qquad (5)\\ \text{s.t.} \quad & \mathbf{X}\in[0,1]^{m_p\times m_q}, \mathbf{X}^T\mathbf{1}=\mathbf{1},\\ & \mathbf{R}\mathcal{Q}(\mathbf{TWS}^T)+\mathbf{b1}^T-\mathbf{PX}=\mathbf{U}-\mathbf{V},\ \mathbf{U},\mathbf{V}\geq\mathbf{0},\\ & \mathbf{TW\Sigma}^{-1}=\mathbf{U}_s-\mathbf{V}_s,\ \mathbf{U}_s,\mathbf{V}_s\geq\mathbf{0},\\ & \mathbf{GPX}-\mathbf{G'P'X'}=\mathbf{U}_o-\mathbf{V}_o,\ \mathbf{U}_o,\mathbf{V}_o\geq\mathbf{0},\end{aligned}$$

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:

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

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

$$\mathbf{\Pi}_{\mathcal{Q}}\operatorname{vec}\left(\begin{bmatrix}\mathbf{q}_1^1 & \cdots & \mathbf{q}_{m_q}^1\\ \vdots & \ddots & \vdots\\ \mathbf{q}_1^n & \cdots & \mathbf{q}_{m_q}^n\end{bmatrix}\right)=\operatorname{vec}\left(\begin{bmatrix}\mathbf{q}_1^{1T} & \cdots & \mathbf{q}_{m_q}^{1T}\\ \vdots & \ddots & \vdots\\ \mathbf{q}_1^{nT} & \cdots & \mathbf{q}_{m_q}^{nT}\end{bmatrix}\right).$$

After solving the linear program, we gradually discretized $\mathbf{X}$ by taking successive refinements based on trust-region shrinking [49]. More specifically, we first initialized a candidate trajectory set for each joint $c\in 1\cdots m_q$ as well as all of the original trajectories $\mathbf{P}=[\mathbf{p}_i]_{i=1}^{m_p}$. After computing $\mathbf{X}$ in the first iteration, we calculated the distance between each candidate trajectory $\mathbf{p}_i$ and the reconstructed trajectory $\mathbf{Px}_c$. We then shrink the candidate set by discarding those trajectories with larger distance. During the second iteration to compute $\mathbf{X}$, we added a new linear constraint to enforce that the non-zero elements in $\mathbf{X}$ can only correspond to the trajectories remaining in the candidate set. After each iteration, we shrink the candidate trajectory set by half until we have only one candidate trajectory for each joint.

6.3 Optimizing STM over R and b

If $\mathbf{X}$ and $\mathbf{W}$ are fixed, optimizing Eq. 4 with respect to $\mathbf{R}$ and $\mathbf{b}$ becomes an l_1 Procrustes problem [50],

$$\min_{\mathbf{R},\mathbf{b}} \quad \|\mathbf{RQ}+\mathbf{b1}^T-\mathbf{PX}\|_1, \quad \text{s.t. } \mathbf{R}\in\Psi, \qquad (6)$$

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 [51] that minimizes the following augmented Lagrange function:

$$\begin{aligned}\min_{\mathbf{L},\mathbf{E},\mu,\mathbf{R},\mathbf{b}} \ & \|\mathbf{E}-\mathbf{PX}\|_1+\operatorname{tr}\left(\mathbf{L}^T(\mathbf{RQ}+\mathbf{b1}^T-\mathbf{E})\right)\\ & +\frac{\mu}{2}\|\mathbf{RQ}+\mathbf{b1}^T-\mathbf{E}\|_F^2, \quad \text{s.t. } \mathbf{R}\in\Psi, \qquad (7)\end{aligned}$$

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,

$$\min_{\mathbf{R},\mathbf{b}} \quad \|\mathbf{RQ}+\mathbf{b1}^T-(\mathbf{E}-\frac{\mathbf{L}}{\mu})\|_F^2, \quad \text{s.t. } \mathbf{R}\in\Psi, \qquad (8)$$

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 [51] as,

$$\mathbf{E}=\mathbf{PX}-\mathcal{S}_{\frac{1}{\mu}}(\mathbf{PX}-\mathbf{RQ}-\mathbf{b1}^T-\frac{\mathbf{L}}{\mu}), \qquad (9)$$

where $\mathcal{S}_\sigma(p)=\max(|p|-\sigma,0)\operatorname{sign}(p)$ is a soft-thresholding operator [51]. Third, we gradually update $\mathbf{L}$ and μ as,

$$\begin{aligned}&\mathbf{L}\leftarrow\mathbf{L}+\mu(\mathbf{RQ}+\mathbf{b1}^T-\mathbf{E}), \qquad (10)\\ &\mu\leftarrow\rho\mu, \qquad (11)\end{aligned}$$

where we set the incremental ratio to $\rho=1.05$ in all our experiments.

Overall, the algorithm is summarized in Algorithm. 1.

Algorithm 1: l_1 Procrustes analysis

```
parameter: $\rho = 1.05$
input    : $\mathbf{PX}\in\mathbb{R}^{2n\times m_q}$, $\mathbf{Q}\in\mathbb{R}^{3n\times m_q}$
output   : $\mathbf{R}\in\mathbb{R}^{2n\times 3n}$, $\mathbf{b}\in\mathbb{R}^{2n}$
begin
    Initialize $\mathbf{E}=\mathbf{0}_{2n\times m_q}$, $\mathbf{L}=\mathbf{0}_{2n\times m_q}$, $\mu=1e-6$ ;
    while not converged do Outer Iteration
        while not converged do Inner Iteration
            Updating $\mathbf{R}$ and $\mathbf{b}$ by optimizing Eq. 8 using SVD;
            Updating $\mathbf{E}$ using Eq. 9;
        Updating $\mathbf{L}$ using Eq. 10;
        Updating $\mu$ using Eq. 11;
```

6.4 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. 4b) in the training step. To find the best model for each segment, we optimize Eq. 4 using each model and select the one with the smallest objective.

After solving STM for each segment, we need to aggregate the local solutions for each segment to generate the global one for the entire sequence. Specifically, how to generate the coordinate $\bar{\mathbf{P}}^i\in\mathbb{R}^{2\times m_q}$ at i^{th} frame from the selected trajectories $\{\mathbf{P}_c\mathbf{X}_c\}_c$ of l_c segments overlapped at i^{th} frame. In the following, we explore two ways to compute $\bar{\mathbf{P}}^i$.

[5] Given an LP problem with absolute value in the objective, *e.g.*, $\min_x |x|$, we can equivalently solve, $\min_{x,u,v} u+v$, by introducing two positive auxiliary variables $u,v\geq 0$ with the constraint $x=u-v$.

Averaging. The first solution is to average the coordinates of the selected trajectories $\{\mathbf{P}_c\mathbf{X}_c\}_c$ overlapped at i,

$$\bar{\mathbf{P}}^i = \frac{1}{l_c}\sum_c \mathbf{P}_c^{i_c}\mathbf{X}_c, \tag{12}$$

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.

Winner-Take-All. In the second way, we evaluate the objective value (Eq. 4) of each local solution $\{\mathbf{P}_c\mathbf{X}_c\}_c$ marginalized at the overlapped frame i, *i.e.*,

$$\begin{aligned} J_c = &\|\big[\mathbf{R}\mathcal{Q}(\mathbf{TWS}^T) + \mathbf{b}\mathbf{1}^T - \mathbf{PX}\big]_{i_c}\|_1 + \lambda_a\|\big[\mathbf{AX}\big]_{i_c}\|_1 \\ &+ \lambda_s\|\big[\mathbf{TW}\mathbf{\Sigma}^{-1}\big]_{i_c}\|_1, \end{aligned} \tag{13}$$

where the operator $[\cdot]_{i_c}$ is to take the rows of a matrix associated to i_c^{th} frame. Once the J_cs for each segment are computed, we pick the one $c^* = \arg\min_c J_c$ with the minimum objective value as the final solution $\bar{\mathbf{P}}^i = \mathbf{P}_{c^*}^{i_c^*}\mathbf{X}_{c^*}$.

7 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 [42], and real experiments on the MHAD [9], the Human3.6M [10] and the MAD [11] 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. 3a). 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. In addition to the different choices of 3D models, we also testified the effect of using different fusion methods. The postfix notations, **-AVE** and **-WTA** stand for using the averaging and winner-take-all methods in the fusion step respectively.

To evaluate the performance of different methods, we adopted the PCK [5] criteria, a popular criteria for evaluating pose estimation accuracy. PCK measures the percentage of correctly localized body parts. More specifically, PCK defines a candidate key-point $\mathbf{p}$ to be correct if it falls within a small range of the ground-truth keypoint $\bar{\mathbf{p}}$, *i.e.*,

$$\|\mathbf{p} - \bar{\mathbf{p}}\|_2 \leq \alpha \max(h, w), \tag{14}$$

where h and w are the height and width of the bounding box respectively, and α controls the relative threshold for considering correctness. Similar to the full-body case in [5], we choose $\alpha = 0.2$ in the experiments. For each frame, we generated the bounding box as the tightest one to cover the set of ground truth key-points.

7.1 CMU motion capture dataset

The first experiment validated our approach on the CMU motion capture dataset [42], 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. The first 3D point in each trajectory was generated randomly, and the subsequent 3D points are shifted with a random vector $\delta\mathbf{p} \sim \beta\mathcal{N}(\mu, \mathbf{I})$ at each frame, where $\mu \in \mathbb{R}^3$ denotes the size of the 3D bounding box of the person and $\beta = [0.01, 0.05]$ controls the velocity of random translation. Then we projected each sequence (with outliers included) onto 4 different 2D views. See Fig. 6a 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. 6b shows some key-frames for the greedy approach, our method and the ground truth using the STM-A4-AVE 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. 7a-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-AVE model obtains lower error rates than STM-G1-AVE because STM-A1-AVE is an action-specific model, unlike STM-G1-AVE which is a generic one. By increasing the number of clusters from one to four, the performance of STM-G4-AVE and STM-A4-AVE clearly improves from STM-G1-AVE and STM-A1-AVE 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

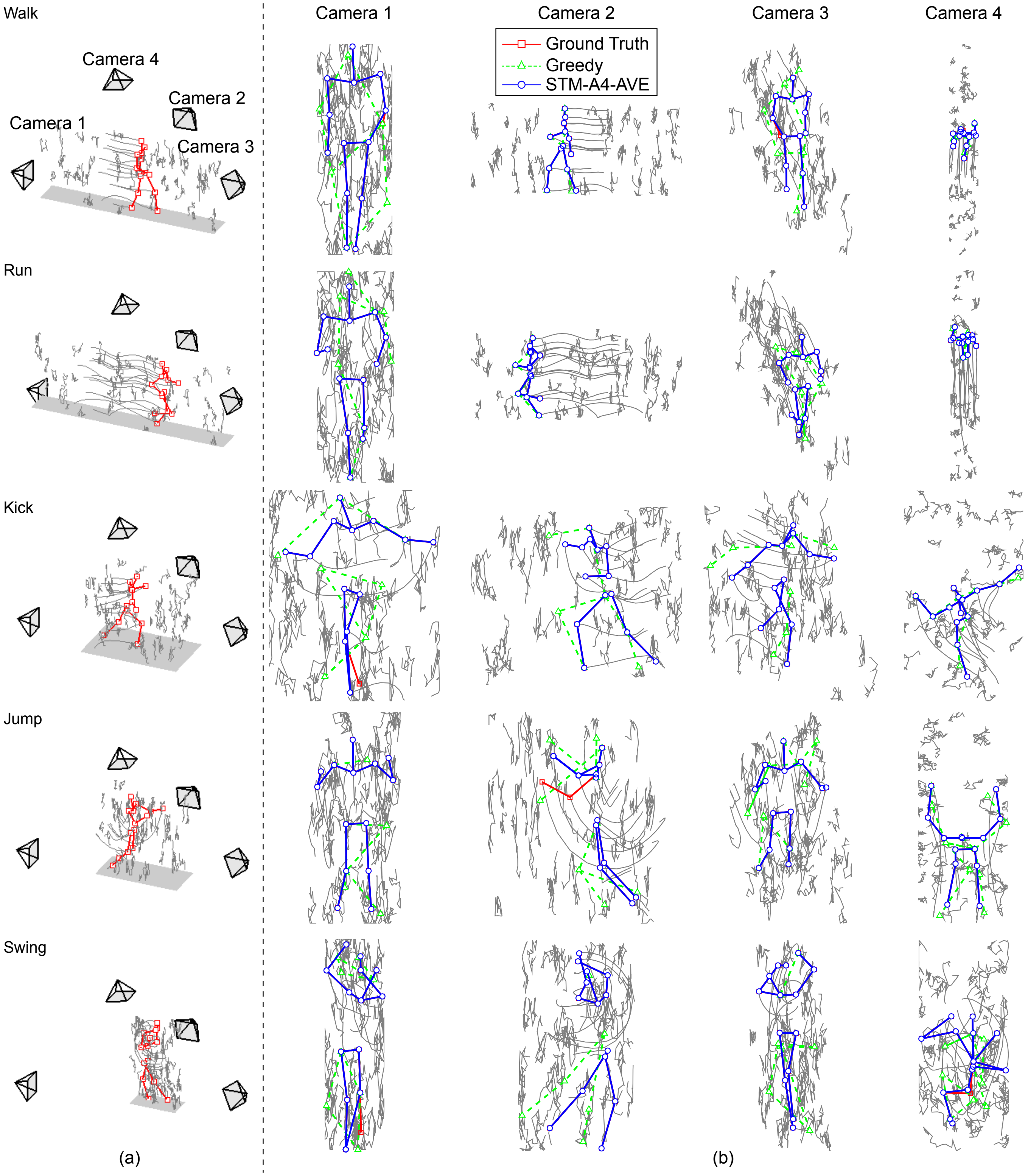


Fig. 6. Comparison of human pose estimation on the CMU motion capture dataset. (a) Original motion capture key-frames in 3D with 50 ($\beta = 0.04$) outliers that were synthetically generated. (b) Results of the greedy approach and our method on four 2D projections.

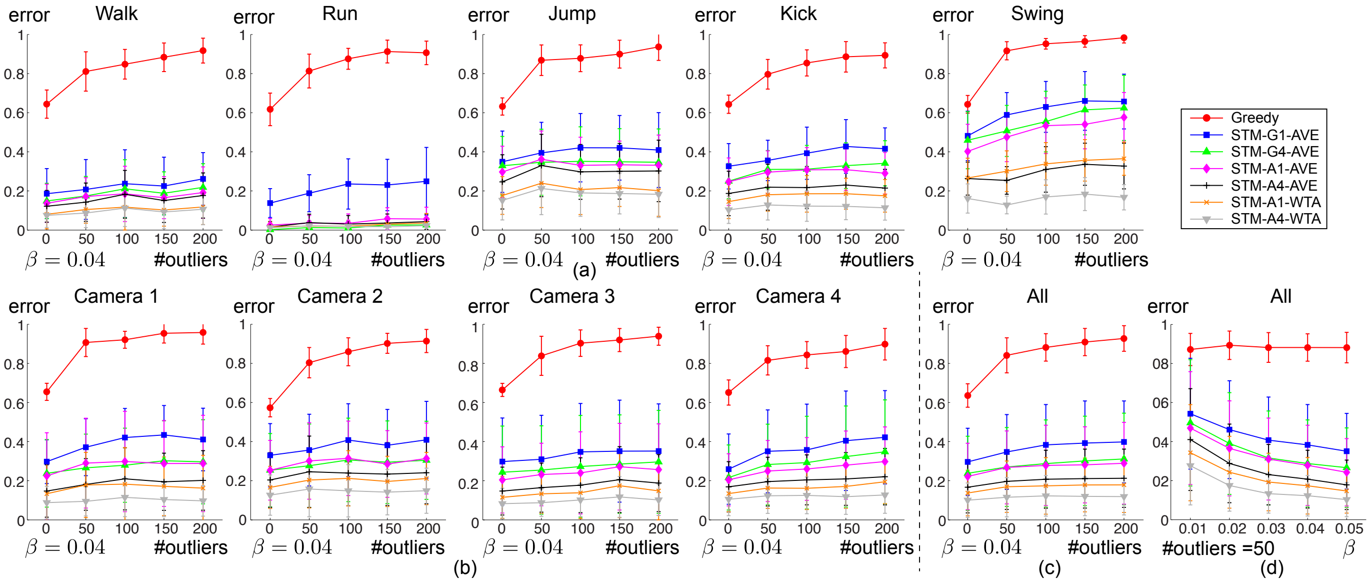


Fig. 7. Comparison of human pose estimation on the CMU motion capture dataset. (a) Mean error and std. for each method and each action as a function of the number of outliers when $\beta = 0.04$. (b) Mean error and std for each camera view when $\beta = 0.04$. (c) Mean error and std for all actions and cameras when $\beta = 0.04$. (d) Mean error and std for all actions and cameras as a function of the velocity (β) of 50 random outliers.

better in testing. In addition, using the winner-take-all method (STM-A1-WTA and STM-A4-WTA) in the fusion step can further improve the performance compared to the averaging approaches (STM-A1-AVE and STM-A4-AVE). This is because the winner-take-all method picks the best local solution at the overlapped region, yielding more robust solution in the case when a large number of outliers exist.

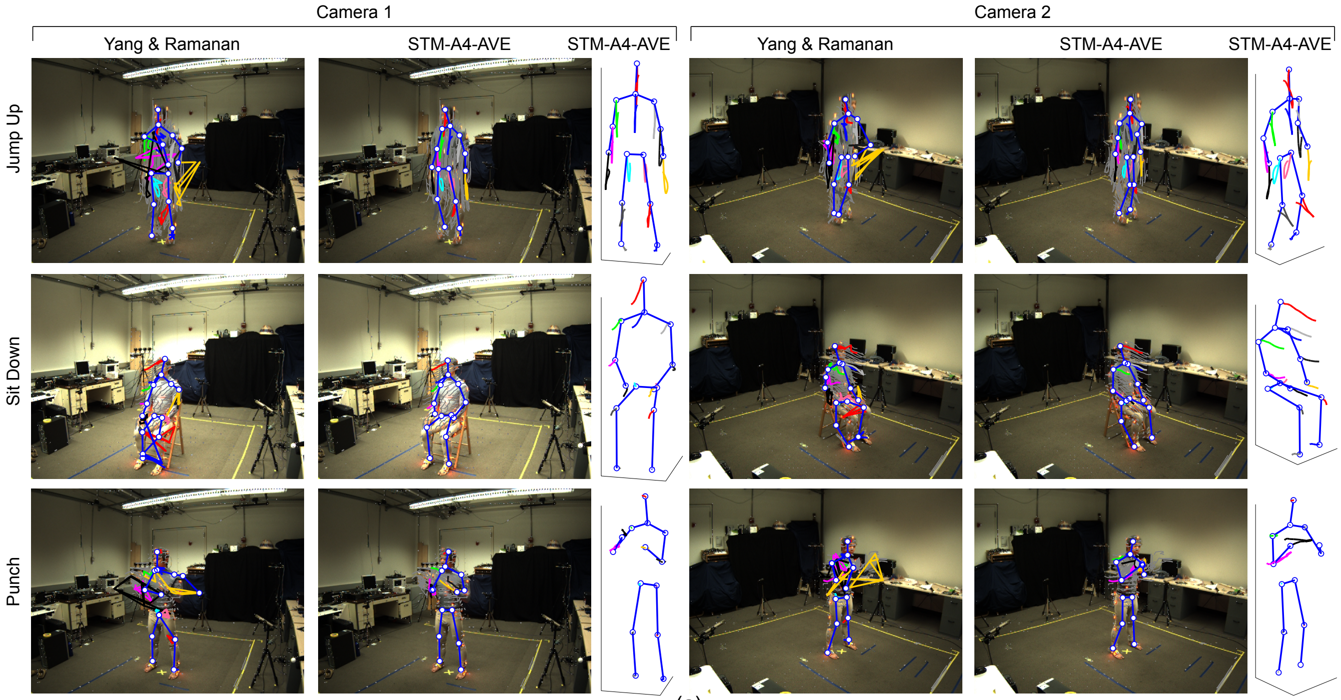
7.2 Berkeley multi-modal human action dataset (MHAD)

In the second experiment, we tested the ability of STM to detect humans on the Berkeley multi-modal human action database (MHAD) [9]. 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. 8a. To extract the trajectories from each video, we used [35] 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 two recent video-based methods designed by Park and Ramanan [25] and Burgos *et al.* [24] that merge 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. 8b-d compare the PCK accuracy score [5] to localize body parts of our method against [5], [24], [25]. The PCK score is computed by setting the threshold $\alpha = 0.2$ in Eq. 14. Our method largely improves the image-based baseline [5] for all actions and viewpoints. Compared to the video-based method [24], STM achieves higher accuracy 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-AVE performs the best because the clustering step improves the generalization of the bilinear model. In this experiment, however, we found taking the average coordinates of the local solutions in fusion steps (STM-A1-AVE and STM-A4-AVE) yielded higher accuracy than the ones (STM-A1-WTA and STM-A4-WTA) using the winner-take-all mechanism. This is because the noise (*e.g.*, drifting and missing point) generated in the dense tracking step can be mitigated by the averaging step. As shown in Fig. 8d, the hands are the most difficult to accurately detect because of their fast movements and frequent occlusions. Fig. 8e compares the PCK scores by adjusting the threshold parameter α in Eq. 14 from 0.1 to 1. Our method consistently out-performed the baselines [5], [24], [25].

Fig. 9 investigates 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. 9a, a smaller segment length is beneficial for "jump jacking" because the performance of the tracker [35] is less stable for fast-speed action. In contrast, using



(a)

$\alpha = 0.2$ PCK (%)	Jump Up	Jump Jack	Bend	Punch	Two-hand Wave	One-hand Wave	Clap	Throw	Sit Down Stand Up	Sit Down	Stand Up	All
Yang & Ramanan	95.6	94.4	66.1	89.8	94.4	96.3	91.5	86.3	87.7	83.9	87.3	88.5
Yang & Park	97.4	**95.8**	67.2	90.6	**95.6**	97.1	92.2	87.5	89.0	85.2	88.7	89.7
Burgos et al.	96.9	94.3	**68.9**	92.0	95.3	97.1	94.6	86.9	91.6	88.9	92.0	90.8
STM-G1-AVE	97.8	91.3	65.2	94.3	93.2	96.5	94.8	91.7	93.7	89.3	94.5	91.1
STM-G4-AVE	98.5	90.9	65.2	94.3	92.7	96.6	**94.9**	92.4	95.1	91.3	95.5	91.6
STM-A1-AVE	**99.8**	91.1	65.1	95.8	91.7	97.7	94.6	92.7	97.9	94.1	99.0	92.7
STM-A4-AVE	99.7	91.5	66.2	**95.9**	93.1	**98.1**	94.4	**92.8**	**98.6**	**95.6**	**99.3**	**93.2**
STM-A1-WTA	99.5	91.7	66.3	94.9	92.6	97.3	94.2	92.2	96.6	93.2	97.8	92.4
STM-A4-WTA	99.6	91.9	67.2	95.2	94.0	97.8	93.9	92.3	97.3	94.6	98.1	92.9

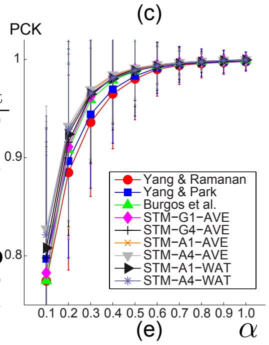
(b)

Cam1	Cam2
91.4	85.6
92.4	87.0
92.8	88.8
92.8	89.4
93.2	90.0
94.1	91.3
94.5	**92.0**
93.8	91.0
94.3	91.5

(c)

$\alpha = 0.2$ PCK (%)	Head	Neck	Right Shou.	Right Elbow	Right Hand	Right Hip	Right Knee	Right Foot	Left Shou.	Left Elbow	Left Hand	Left Hip	Left Knee	Left Foot
Yang & Ramanan	93.5	95.2	95.3	82.0	59.5	93.6	91.5	94.3	95.4	84.8	67.2	94.5	96.0	96.0
Yang & Park	93.6	95.4	95.4	84.2	63.4	94.1	92.2	95.1	95.7	86.6	70.5	95.0	96.9	97.2
Burgos et al.	92.9	96.0	96.0	86.5	64.1	95.8	94.5	94.6	96.1	89.4	74.3	95.9	98.2	96.8
STM-G1-AVE	**96.0**	97.3	**97.4**	87.0	54.8	97.7	97.2	96.3	**97.4**	90.4	68.7	98.2	99.1	98.1
STM-G4-AVE	**96.0**	**97.4**	**97.4**	87.4	57.7	97.6	**97.6**	97.8	**97.4**	89.5	70.1	98.3	99.3	98.5
STM-A1-AVE	95.6	97.2	97.3	88.8	67.8	98.2	97.2	96.9	97.3	91.0	74.7	98.5	99.1	98.1
STM-A4-AVE	95.8	97.2	97.3	**89.2**	69.6	**98.4**	97.4	**97.9**	**97.4**	**91.2**	76.5	**98.7**	**99.4**	**98.9**
STM-A1-WTA	95.9	96.8	96.7	88.2	67.7	97.8	96.2	97.3	97.0	90.5	74.9	98.0	98.4	98.0
STM-A4-WTA	**96.0**	96.9	96.8	88.4	**69.7**	97.9	96.7	97.7	97.1	90.6	**76.7**	98.2	98.8	98.8

(d)



(e)

Fig. 8. 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. Videos are available at `www.f-zhou.com/hpe/fig8a_vdo1.avi`, `www.f-zhou.com/hpe/fig8a_vdo2.avi` and `www.f-zhou.com/hpe/fig8a_vdo3.avi`. (b) PCK accuracy for each action. (c) PCK accuracy for each camera view. (d) PCK accuracy of each joint. (e) PCK as a function of threshold α

a larger window improves the temporal consistency in actions such as "throwing" and "standing up". Fig. 9b shows the detection error (pixel distance) 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. 4) 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. 8c 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], [24] were downloaded from authors' webpages. The linear programming in Eq. 5 was optimized using the Mosek LP solver [52]. Fig. 9d analyzes the computational cost

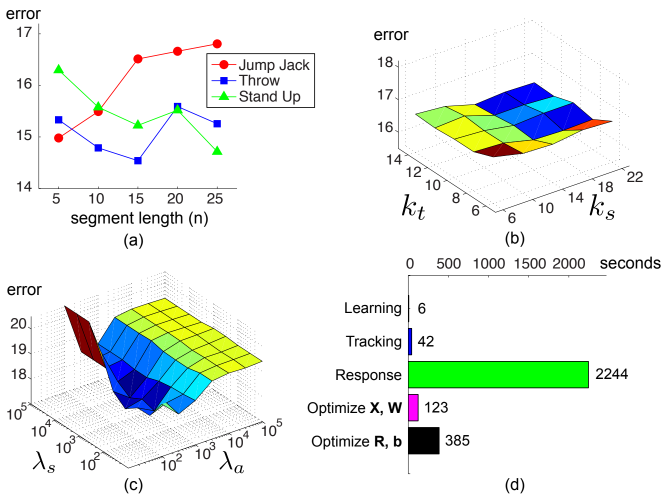


Fig. 9. Comparison of human pose estimation on the Berkeley MHAD dataset. (a) Errors with respect to the segment length (n). (b) Errors with respect to the bases number (k_s and k_t). (c) Errors with respect to the regularization weights (λ_a and λ_s). (d) Time cost of each step.

(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.

7.3 Human3.6M dataset

In this experiment, we selected 15 actions performed by 5 subjects from the Human3.6M dataset [10]. 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. 10a for example frames.

As in the previous experiment, our methods were compared with three baselines [5], [24], [25] in a leave-one-out scheme. The bilinear models were trained from the motion capture data associated with this dataset. Fig. 8b-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 PCK accuracy of each method is lower than the one achieved on the previous Berkeley MHAD dataset. However, STM still outperforms both the baselines [5], [24], [25] 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. 8d shows the PCK score for each joint. Among the 14 joints, hands and elbows are the most difficult to track accurately because of their large movement relative to the body. Fig. 8e compares the overall PCK scores using different threshold value α. The proposed STM-A4-AVE consistently achieved the best performance.

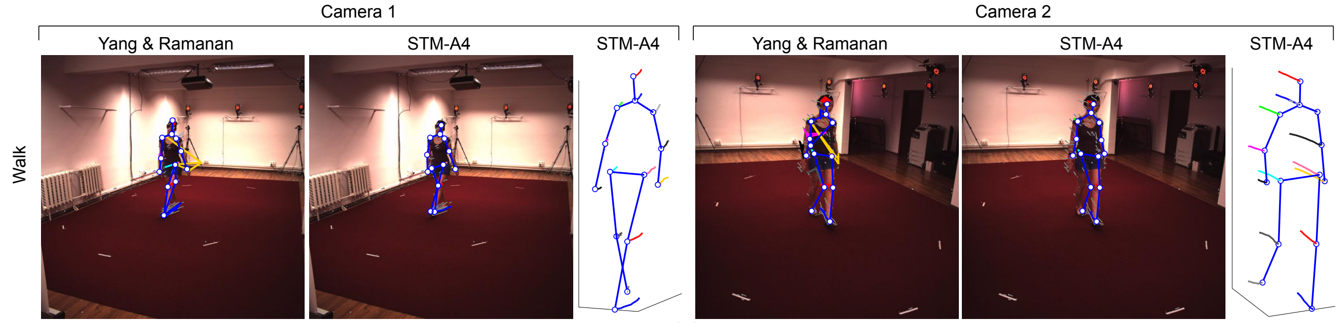
7.4 CMU multi-modal action dataset (MAD)

In the last experiment, we tested our method on the CMU multi-modal action detection (MAD) dataset [11]. Unlike MHAD and H3M datasets where each sequence contains only one action, MAD contains 40 sequences of 20 subjects (2 sequences per subject) performing 35 activities in each of the sequences. Therefore, this experiment can evaluate more accurately the performance of different human pose estimation methods in real applications. The length of each sequence is around $2-4$ minutes ($4000-7000$ frames). The 2D and 3D coordinates of 14 joints for each frame was recorded using the Microsoft Kinect sensor in an indoor environment. The 35 actions include full-body motion (*e.g.*, run, crouch, jump), upper-body motion (*e.g.*, throw, basketball dribble, baseball swing), and lower-body motion (*e.g.*, kick). Each subject performs all the 35 activities continuously, and the segments between two actions are considered the null class (*i.e.*, the subject is standing). Fig. 11a show some example frames (bottom) and frame labels (top) for two sequences.

As in the previous experiment, our methods were compared with three baselines [5], [24], [25] in a leave-one-subject-out scheme. The bilinear models were trained from the Kinect data associated with this dataset. The middle line of Fig. 11a compares the PCK score between the baseline [5] and the proposed STM-A4-AVE for each frame. STM-A4-AVE is able to locate the body parts more smoothly and accurately over time. Fig. 11b shows the performance of each method in the task of localizing body parts. Fig. 11c shows the PCK accuracy of each joint. Compared to the baselines [5], [24], [25], STM achieved higher accuracy in most joints, especially the hand and elbow which had larger movements. If the action label is known a priori, training action-specific models (STM-A1-AVE and STM-A4-AVE) achieves better performance than the ones trained on all actions (STM-G1-AVE and STM-G4-AVE). Similar to previous results, the averaging fusion step (STM-A1-AVE and STM-A4-AVE) performed better than the winner-take-all (STM-A1-WTA and STM-A4-WTA). Fig. 11d evaluates the PCK scores at different levels of α. Our methods (STM-A1-AVE and STM-A4-AVE) out-performed the baselines [5], [24], [25] by a large margin..

8 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



(a)

$\alpha = 0.2$ PCK (%)	Direct	Discuss	Eat	Greet	Phone Talk	Pose	Buy	Sit	Sit Down	Smoke	Take Photo	Wait	Walk	Walk Dog	Walk Pair	All
Yang & Ramanan	63.4	60.6	50.9	64.8	51.5	55.9	46.8	39.2	25.2	53.1	52.4	62.7	75.6	57.2	74.9	55.6
Yang & Park	67.1	**62.7**	52.2	66.8	53.0	**58.0**	46.7	39.9	25.9	54.5	53.2	63.7	77.3	59.1	76.3	57.1
Burgos et al.	65.6	57.3	53.1	65.9	52.8	56.0	43.8	40.0	24.0	53.5	51.3	59.9	76.8	56.7	75.3	55.5
STM-G1-AVE	66.7	62.3	54.1	68.5	55.9	57.3	47.6	41.9	26.6	57.8	54.9	65.3	80.7	60.8	79.3	58.6
STM-G4-AVE	67.1	62.4	54.2	68.8	56.1	57.7	47.5	41.7	26.2	58.1	**55.0**	65.2	81.0	**61.2**	**79.6**	58.8
STM-A1-AVE	66.9	62.1	54.8	68.3	56.4	57.3	**47.7**	42.2	26.3	58.3	54.7	65.6	81.4	61.0	79.4	58.8
STM-A4-AVE	**67.6**	62.3	**55.1**	**68.9**	**56.5**	57.6	**47.7**	**42.4**	26.2	**58.7**	54.9	**65.9**	**81.5**	**61.2**	79.4	**59.1**
STM-A1-WTA	66.1	61.2	54.2	67.2	55.6	56.6	47.0	41.7	**26.7**	57.6	54.2	64.7	80.1	60.4	78.4	58.1
STM-A4-WTA	66.5	61.3	54.4	67.7	55.7	57.0	47.2	42.0	**26.7**	57.9	54.2	65.0	80.3	60.5	78.3	58.3

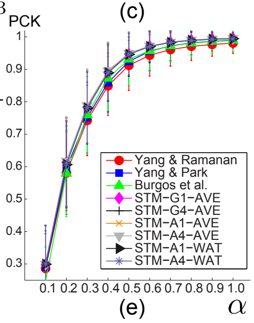
(b)

Cam1	Cam2
56.2	55.1
58.0	56.1
56.1	54.8
58.9	58.4
59.0	58.6
59.0	58.6
59.2	**59.0**
58.4	57.9
58.5	58.2

(c)

$\alpha = 0.2$ PCK (%)	Head	Neck	Right Shou.	Right Elbow	Right Hand	Right Hip	Right Knee	Right Foot	Left Shou.	Left Elbow	Left Hand	Left Hip	Left Knee	Left Foot
Yang & Ramanan	86.8	89.6	60.3	21.4	22.4	75.6	62.1	58.1	63.6	22.7	21.9	75.2	63.7	55.5
Yang & Park	91.1	91.5	58.3	**21.5**	22.5	**79.0**	**65.4**	62.5	61.4	21.8	21.4	78.5	65.2	59.1
Burgos et al.	83.1	87.3	55.7	21.0	22.6	**79.0**	65.0	**62.6**	56.8	22.3	21.7	78.3	64.7	56.5
STM-G1-AVE	93.1	94.5	**64.4**	21.3	26.1	78.3	64.4	60.1	**67.6**	22.7	24.8	78.4	67.1	58.3
STM-G4-AVE	93.3	**94.6**	64.2	21.1	25.7	78.3	64.3	61.9	67.4	22.6	24.9	78.2	67.3	59.2
STM-A1-AVE	93.1	**94.6**	64.2	21.2	26.4	78.3	64.4	60.8	67.5	**22.8**	25.9	**78.6**	67.0	58.7
STM-A4-AVE	93.3	**94.6**	63.9	21.0	**26.5**	78.1	64.6	62.5	67.3	22.6	**26.6**	**78.6**	**67.4**	**59.7**
STM-A1-WTA	93.2	94.5	63.1	21.1	26.0	76.7	63.5	60.5	66.1	22.3	25.5	77.0	65.8	58.2
STM-A4-WTA	**93.4**	94.5	62.9	20.9	26.0	76.6	63.8	61.7	66.1	22.3	26.0	77.2	66.2	58.9

(d)



(e)

Fig. 10. 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) PCK accuracy for each action. (c) PCK accuracy for each camera view. (d) PCK accuracy of each joint. (e) PCK as a function of threshold α

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 Berkeley MHAD [9], the Human3.6M [10] and the CMU MAD [11] dataset.

A major limitation of our current approach is the high computational cost for calculating the joints' response, which is computed independently for each frame. In future work, we plan to incorporate richer temporal features [35] to improve the speed and accuracy of the trajectory response.

References

[1] J. Shotton, R. B. Girshick, A. W. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, "Efficient human pose estimation from single depth images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2821–2840, 2013.

[2] O. Boiman and M. Irani, "Detecting irregularities in images and in video," *Int. J. Comput. Vis.*, vol. 74, no. 1, pp. 17–31, 2007.

[3] X. K. Wei and J. Chai, "VideoMocap: modeling physically realistic human motion from monocular video sequences," *ACM Trans. Graph.*, vol. 29, no. 4, 2010.

[4] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.

[5] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2878–2890, 2013.

[6] M. Andriluka, S. Roth, and B. Schiele, "Discriminative appearance models for pictorial structures," *Int. J. Comput. Vis.*, vol. 99, no. 3, pp. 259–280, 2012.

[7] C. Ionescu, F. Li, and C. Sminchisescu, "Latent structured models for human pose estimation," in *ICCV*, 2011.

[8] M. Eichner, M. Jesús, A. Zisserman, and V. Ferrari, "2d articulated human pose estimation and retrieval in (almost) unconstrained still images," *Int. J. Comput. Vis.*, vol. 99, no. 2, pp. 190–214, 2012.

[9] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley MHAD: A comprehensive multimodal human action database," in *IEEE Workshop on Applications on Computer Vision (WACV)*, 2013, pp. 53–60.

[10] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014.

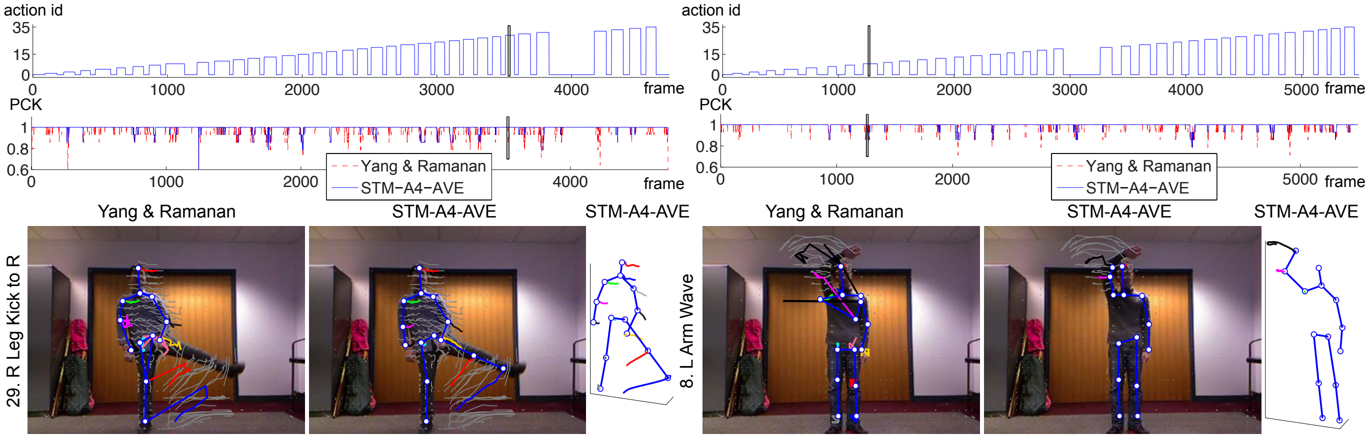
[11] D. Huang, Y. Wang, S. Yao, and F. De la Torre, "Sequential max-margin event detectors," in *ECCV*, 2014.

[12] S. X. Ju, M. J. Black, and Y. Yacoob, "Cardboard people: A parameterized model of articulated image motion," in *FG*, 1996.

[13] J. Deutscher and I. D. Reid, "Articulated body motion capture by stochastic search," *Int. J. Comput. Vis.*, vol. 61, no. 2, pp. 185–205, 2005.

[14] C. Bregler and J. Malik, "Tracking people with twists and exponential maps," in *CVPR*, 1998.

[15] K. Rohr, "Incremental recognition of pedestrians from image sequences," in *CVPR*, 1993.



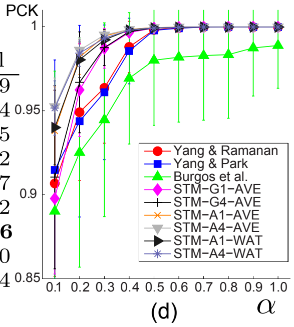
(a)

$\alpha = 0.2$ PCK (%)	Run	Crouch	Jump	Walk	Jump Side-Kick	L Arm Swipe to L	L Arm Swipe to R	L Arm Wave	L Arm Punch	L Arm Dribble	L Arm Point	L Arm Throw	L Arm Swing	L Arm Receive
Yang & Ramanan	96.6	95.1	96.7	96.1	96.5	94.8	94.5	94.4	93.9	95.8	93.3	94.6	92.0	94.2
Yang & Park	96.8	96.2	96.7	96.1	96.3	93.3	93.5	94.9	93.9	95.5	93.8	94.5	90.8	93.7
Burgos et al.	95.5	95.9	96.2	95.6	96.3	93.4	93.4	93.7	93.0	95.3	91.9	94.4	88.4	93.8
STM-G1-AVE	98.4	97.2	98.9	96.9	97.9	96.2	95.5	94.9	96.9	97.2	94.2	96.6	93.6	96.2
STM-G4-AVE	98.7	98.1	98.3	97.7	97.8	96.0	95.8	96.2	96.0	98.0	95.1	96.4	93.9	96.4
STM-A1-AVE	99.6	98.8	99.6	99.8	99.1	97.8	98.0	96.7	**97.8**	99.2	95.4	97.6	95.3	97.6
STM-A4-AVE	**99.8**	**99.2**	**100.0**	**100.0**	**99.4**	97.3	97.7	**97.1**	**97.8**	**99.7**	**96.2**	**97.9**	**95.7**	**97.9**
STM-A1-WTA	99.4	98.4	99.4	99.4	98.9	**97.9**	**98.2**	96.3	97.6	99.0	95.1	96.9	94.4	97.6
STM-A4-WTA	99.6	98.6	99.8	**100.0**	99.2	97.5	97.9	97.0	97.4	99.4	95.8	97.5	95.1	**97.9**

$\alpha = 0.2$ PCK (%)	L Arm Back Receive	L Leg Kick to Front	L Leg Kick to L	R Arm Swipe to L	R Arm Swipe to R	R Arm Wave	R Arm Punch	R Arm Dribble	R Arm Point	R Arm Throw	Swing from R	R Arm Receive	R Arm Back Receive
Yang & Ramanan	93.3	95.6	94.9	94.6	94.9	94.5	94.9	95.5	93.8	94.6	91.4	94.4	93.7
Yang & Park	92.3	94.6	94.0	93.4	93.2	94.8	94.1	93.7	94.1	94.0	91.0	94.1	92.7
Burgos et al.	92.9	95.2	93.8	93.3	93.7	93.9	94.3	94.6	93.1	93.7	90.8	93.8	92.8
STM-G1-AVE	94.6	98.5	96.3	95.3	96.5	94.4	96.5	96.5	91.9	95.0	92.0	95.7	94.1
STM-G4-AVE	94.9	98.4	98.1	96.0	96.8	96.0	96.9	97.7	93.5	96.2	92.4	96.9	94.2
STM-A1-AVE	95.2	**100.0**	**99.5**	**98.9**	**99.2**	95.9	96.7	99.6	95.4	96.4	93.7	98.1	95.9
STM-A4-AVE	**95.6**	**100.0**	**99.5**	**98.9**	98.7	**98.4**	**97.7**	**99.7**	**96.8**	**98.2**	**94.2**	**98.6**	**97.1**
STM-A1-WTA	94.9	99.7	99.3	98.4	98.8	97.0	96.4	99.4	96.2	95.9	93.8	97.9	95.7
STM-A4-WTA	95.5	99.8	99.3	98.5	98.5	98.2	96.9	**99.7**	96.4	97.6	94.1	98.5	96.5

$\alpha = 0.2$ PCK (%)	R Leg Kick to Front	R Leg Kick to R	C Arms in Chest	Basket-ball Shoot	B Arms Point to Screen	B Arms Point to B Sides	B Arms Point to R Side	B Arms Point to L Side	Null Class	All
Yang & Ramanan	95.3	94.5	91.9	92.5	94.3	97.5	93.9	95.8	95.5	94.9
Yang & Park	95.3	94.4	90.7	91.9	94.1	96.9	93.4	95.9	94.7	94.4
Burgos et al.	94.4	93.9	80.4	71.8	77.0	80.3	92.5	93.8	93.3	92.5
STM-G1-AVE	97.2	95.1	94.6	94.1	96.8	94.8	94.6	96.9	97.3	96.2
STM-G4-AVE	97.9	96.0	92.9	94.2	97.1	98.3	95.0	96.6	97.5	96.7
STM-A1-AVE	98.9	97.5	**95.9**	95.1	98.7	97.4	95.9	97.9	99.6	98.2
STM-A4-AVE	**99.3**	**99.1**	95.8	**95.2**	**99.3**	**98.5**	**96.3**	**98.0**	**99.9**	**98.6**
STM-A1-WTA	98.5	97.4	95.6	94.9	98.2	96.9	95.3	97.7	99.5	98.0
STM-A4-WTA	99.1	98.6	95.7	94.7	98.8	98.2	95.2	97.9	99.8	98.4

(b)



(d)

$\alpha = 0.2$ PCK (%)	Head	Neck	Right Shoulder	Right Elbow	Right Hand	Right Hip	Right Knee	Right Foot	Left Shoulder	Left Elbow	Left Hand	Left Hip	Left Knee	Left Foot
Yang & Ramanan	**100.0**	**100.0**	99.9	93.7	63.1	99.9	99.6	99.6	99.9	96.6	77.3	99.9	99.7	99.6
Yang & Park	**100.0**	**100.0**	**100.0**	84.6	62.8	**100.0**	99.8	99.7	**100.0**	95.9	79.4	99.9	99.7	99.7
Burgos et al.	98.0	98.0	97.9	83.0	63.0	98.2	97.9	97.8	97.9	94.0	75.8	98.1	97.8	97.8
STM-G1-AVE	**100.0**	**100.0**	**100.0**	96.2	70.1	**100.0**	**99.9**	99.6	**100.0**	98.5	83.7	**100.0**	**99.9**	99.6
STM-G4-AVE	**100.0**	**100.0**	**100.0**	95.1	74.7	**100.0**	**99.9**	99.8	**100.0**	98.6	86.1	**100.0**	**99.9**	**99.8**
STM-A1-AVE	**100.0**	**100.0**	**100.0**	**97.4**	87.4	**100.0**	**99.9**	99.5	**100.0**	98.5	93.2	**100.0**	99.8	99.3
STM-A4-AVE	**100.0**	**100.0**	**100.0**	97.2	**90.4**	**100.0**	**99.9**	**99.9**	**100.0**	**98.8**	**94.8**	**100.0**	**99.9**	**99.8**
STM-A1-WTA	**100.0**	**100.0**	**100.0**	96.9	86.2	**100.0**	99.8	99.7	**100.0**	98.3	92.5	**100.0**	99.8	99.5
STM-A4-WTA	**100.0**	**100.0**	**100.0**	97.0	89.9	**100.0**	99.8	99.8	**100.0**	98.4	93.8	**100.0**	99.8	99.7

(c)

Fig. 11. Comparison of human pose estimation on the CMU MAD dataset. (a) Top: Action id of two sequences. Middle: PCK for each frame. Bottom: Result of [5] and our method on two actions, where the 3D reconstruction estimated by our method is plotted on the right. Videos are available at `www.f-zhou.com/hpe/fig11a_vdo1.avi` and `www.f-zhou.com/hpe/fig11a_vdo2.avi`. (b) PCK accuracy for each action. (c) PCK accuracy of each joint. (d) PCK as a function of threshold α.

[16] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3d human figures using 2d image motion," in *ECCV*, 2000.

[17] J. Deutscher, A. Blake, and I. D. Reid, "Articulated body motion capture by annealed particle filtering," in *CVPR*, 2000.

[18] Y. Song, L. Goncalves, and P. Perona, "Unsupervised learning of human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 814–827, 2003.

[19] B. Sapp, A. Toshev, and B. Taskar, "Cascaded models for articulated pose estimation," in *ECCV*, 2010.

[20] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *CVPR*, 2008.

[21] ——, "Monocular 3d pose estimation and tracking by detection," in *CVPR*, 2010.

[22] B. Sapp, D. Weiss, and B. Taskar, "Parsing human motion with stretchable models," in *CVPR*, 2011.

[23] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors," *Int. J. Comput. Vis.*, vol. 75, no. 2, pp. 247–266, 2007.

[24] X. Burgos, D. Hall, P. Perona, and P. Dollár, "Merging pose estimates across space and time," in *BMVC*, 2013.

[25] D. Park and D. Ramanan, "N-best maximal decoders for part models," in *ICCV*, 2011.

[26] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *CVPR*, 2014.

[27] A. Jain, J. Tompson, Y. LeCun, and C. Bregler, "Modeep: A deep learning framework using motion features for human pose estimation," 2014.

[28] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 44–58, 2006.

[29] A. M. Elgammal and C.-S. Lee, "Inferring 3d body pose from silhouettes using activity manifold learning," in *CVPR*, 2004.

[30] R. Urtasun, D. J. Fleet, and P. Fua, "3d people tracking with Gaussian process dynamical models," in *CVPR*, 2006.

[31] L. Sigal and M. J. Black, "Predicting 3d people from 2d pictures," in *AMDO*, 2006.

[32] E. Simo-Serra, A. Ramisa, G. Alenyà, C. Torras, and F. Moreno-Noguer, "Single image 3d human pose estimation from noisy observations," in *CVPR*, 2012.

[33] A. Yao, J. Gall, and L. J. V. Gool, "Coupled action recognition and pose estimation from multiple views," *Int. J. Comput. Vis.*, vol. 100, no. 1, pp. 16–37, 2012.

[34] T.-H. Yu, T.-K. Kim, and R. Cipolla, "Unconstrained monocular 3d human pose estimation by action detection and cross-modality regression forest," in *CVPR*, 2013.

[35] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, 2013.

[36] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *SCIA*, 2003.

[37] R. Messing, C. J. Pal, and H. A. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *ICCV*, 2009.

[38] P. Matikainen, M. Hebert, and R. Sukthankar, "Trajectons: Action recognition through the motion analysis of tracked features," in *ICCVW*, 2009.

[39] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, no. 2-3, pp. 107–123, 2005.

[40] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *ICCV VS-PETS*, 2005.

[41] H. Wang, M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009.

[42] Carnegie Mellon University Motion Capture Database, http://mocap.cs.cmu.edu.

[43] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical geometry of non-rigid shapes*. Springer, 2008.

[44] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3d shape from image streams," in *CVPR*, 2000.

[45] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Trajectory space: A dual representation for nonrigid structure from motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1442–1456, 2011.

[46] I. Akhter, T. Simon, S. Khan, I. Matthews, and Y. Sheikh, "Bilinear spatiotemporal basis models," *ACM Trans. Graph.*, vol. 31, no. 2, p. 17, 2012.

[47] J. C. Gower and G. B. Dijksterhuis, *Procrustes problems*. Oxford University Press, 2004.

[48] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *NIPS*, 2001, pp. 849–856.

[49] H. Jiang, M. S. Drew, and Z.-N. Li, "Matching by linear programming and successive convexification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 959–975, 2007.

[50] N. Trendafilov, "On the l_1 Procrustes problem," *Future Generation Computer Systems*, vol. 19, no. 7, pp. 1177–1186, 2004.

[51] Z. Lin, M. Chen, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[52] Mosek, http://www.mosek.com/.

ACKNOWLEDGMENT

This work was partially supported by the National Science Foundation under Grant No. EEEC-0540865 and RI-1116583. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Feng Zhou received the BS degree in computer science from Zhejiang University in 2005, the MS degree in computer science from Shanghai Jiao Tong University in 2008, and the PhD degree in robotics from Carnegie Mellon University in 2014. He is now a researcher in the Media Analytics group at NEC Laboratories America. His research interests include machine learning and computer vision.

Fernando de la Torre is an Associate Research Professor in the Robotics Institute at Carnegie Mellon University. He received his B.Sc. degree in Telecommunications, as well as his M.Sc. and Ph. D degrees in Electronic Engineering from La Salle School of Engineering at Ramon Llull University, Barcelona, Spain in 1994, 1996, and 2002, respectively. His research interests are in the fields of Computer Vision and Machine Learning. Currently, he is directing the Component Analysis Laboratory (http://ca.cs.cmu.edu) and the Human Sensing Laboratory (http://humansensing.cs.cmu.edu) at Carnegie Mellon University. He has over 150 publications in referred journals and conferences. He has organized and co-organized several workshops and has given tutorials at international conferences on the use and extensions of Component Analysis.