Multimed Tools Appl (2017) 76:2399–2439
DOI 10.1007/s11042-015-3186-0

3D human pose regression via robust sparse tensor subspace learning

Jialin Yu[1] · **Jifeng Sun**[1]

Received: 3 July 2015 / Revised: 22 November 2015 / Accepted: 18 December 2015 /
Published online: 16 January 2016

Abstract In this paper, we present a novel algorithm called robust sparse tensor subspace learning (RSTSL) for 3D human pose regression, and it further extends the latest presented tensor learning to a sparse case. A set of interrelated sparse discriminant projection matrices for feature extraction can be obtained by introducing *k*-mode optimization and elastic-net algorithm into the objective function of RSTSL and each non-zero element in each discriminant projection matrix is selected from the most typical variables. Thus, the most important low-dimensional tensor feature (LDTF) that corresponds to a test image (i.e., high-order tensor) is extracted through sparse projection transformation. Moreover, we present a novel regression model called optimal-order support tensor regression (OOSTR) to build a finest mapping function between LDTF and 3D human pose configuration. Extensive simulations are conducted on two human motion databases, HumanEva and Brown databases, experimental results show that our proposed RSTSL can not only weaken the sensitivity to incoherent human motions caused by transient occlusion of cameras, sudden change in human velocity and low-frame rate but also strengthen the robustness to silhouette ambiguity, obstacle occlusion and random noise. All the results have confirmed that our tracking system achieves the most significant performance against the compared the state-of-the-art approaches, especially in the complicated human motion databases with different clustered backgrounds, human movements, clothing-style, illumination and subjects like HumanEva database.

Keywords Human pose estimation · Tensor learning · Sparse representation · Support tensor regression · Sparse projection transformation · Feature extraction

✉ Jialin Yu
yu.jialin@mail.scut.edu.cn

Jifeng Sun
ecjfsun@scut.edu.cn

[1] School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China

1 Introduction

In recent years, 3D human pose estimation has become a prevailing research field due to its numerous computer vision applications, including activity recognition [20], gesture recognition [38] and automated visual surveillance [15] and so on. We are mainly interested in recovering the 3D human poses from the silhouettes of multi-view image or video sequence. Although 3D human pose estimation has been completely developed during the last decades, it still has lots of unsolved and challenging difficulties for a series of reasons. Firstly, directly inferring 3D human pose from 2D visual observation is inherently ambiguous and a 2D human body silhouette can correspond to multiple different 3D human poses due to the loss of depth information [39]. Secondly, it is very difficult to determine the 3D spatial location information of body joint centers only from a 2D human silhouette [1]. In addition, differences in conditions like environmental illumination, human body appearance and clothing styles can result in a great variety of images [9]. Finally, self-occlusion, clustered backgrounds and high-dimensionality of pose state space also make 3D human pose estimation a very challenging task. In the past few years, a lot of approaches have been presented to deal with those difficulties. Nevertheless, no approach has been presented to resolve all the above-mentioned difficulties. The existing 3D human pose estimation algorithms can be roughly divided into three different categories: generative, discriminative and hybrid algorithms.

For generative algorithms, the final pose that can best describe the visual observation is estimated by implementing a stochastic search in the hypothesized pose space, and a generative model needs to be constructed which measures how similar the final pose is to the visual observation. A human pose hypothesis that is most similar to the visual observation will be selected as the final pose output, but these algorithms are too time-consuming and low-efficiency [18, 24]. Given a planar visual observation, it can be associated with more than one 3D human pose hypothesis. Such ambiguities can be slightly overcome under the condition of multi-view scenario (e.g., multi-view image or video sequence).

For discriminative algorithms, a finest mapping between the observed visual feature and the 3D human pose needs to be built by using supervised learning algorithms. Although the discriminative algorithms can efficiently recover the 3D human pose, it may produce unreliable output for a new input when the tracking system is trained by a small amount of samples. Additionally, the relationship between the observed visual feature and the 3D human pose is multimodal and complicated. When the visual observation is represented as the silhouette feature descriptors, a human body silhouette can correspond to multiple 3D human pose hypotheses. The ambiguity can be eliminated by constructing more than one model to build one-to-many mappings between the observed visual feature and the 3D human pose.

For hybrid algorithms, the aforementioned two kinds of algorithms are combined together to finish the task of pose estimation. The generative algorithms are very flexible since they take advantage of prior information of pose space and utilize the 3D human body model to explore the 3D pose space. The discriminative algorithms are very powerful for 3D pose estimation when they are only based on a mapping between the 2D visual observation and the 3D human pose. In the work of Rosales et al. [25], a discriminative mapping function is constructed to map the visual data to a hypothesized pose space. The final pose is recovered by implementing a stochastic search in the pose space, but their research is only used to estimate 3D human pose from a monocular image and cannot be used for video-based 3D human pose tracking. Additionally, those algorithms assume that a hypothesized pose space is always reliable, thus, they cannot handle some abnormal situations when the model produces unreliable hypotheses.

In this paper, we mainly focus on recovering 3D human poses from multi-view image sequence within the discriminative framework.

Generally speaking, a discriminative model involves two major components, i.e., feature representation and mapping function constructing, in which the tracking accuracy highly depends on both the representation of visual data and the choice of regression strategy. Specifically, there are three key challenges we attempt to address in our proposed model: (1) the visual data is inherently ambiguous and uncertain when distinguishing the left- and right-side of a human subject under the condition of self-occlusion; (2) incoherent 3D human movements caused by sudden change in human velocity, low-frame rate and transient occlusion of cameras usually result in losing human targets; (3) the obstacle occlusion and random noise are also disadvantageous for the feature representation. Actually, the higher-order representation of visual data is more effective than conventionally clumsy vector [6]. Inspired by this truth, we also consider an input image as a high-order tensor, by which, the intrinsic structural information of the input data is perfectly maintained. However, the high-order tensor is not suitable for generic regression strategies due to its high-dimensionality, and thus a robust tensor subspace learning algorithm is urgently needed. Regression strategy, in general how to build a mapping between visual feature and human pose configuration, is another important component. Although numerous methods have already been proposed, a more effective and adaptive regression strategy is still required for robust 3D human pose estimation.

In this paper, we propose a novel discriminative framework that considers an image as a high-order tensor rather than a high-dimensional vector. Our goal is to extract LDTF from an input high-order tensor and build an effective mapping between LDTF and 3D human pose configuration. Based on the state-of-the-art methods, we propose a novel algorithm called OOSTR for 3D human pose regression. To the best of our knowledge, it is the first research that constructs the human pose regression model in the framework of supervised tensor subspace learning. In sum, the major contributions of this paper are fourfold as described below:

(1) Our proposed RSTSL handles high-order tensor data and learns multiple sparse tensor subspaces along each tensor mode, which is very different from the conventional approaches that first calculate the vector-based solutions and then convert them to the tensor-based space.
(2) We build a multi-linear mapping between LDTF and pose configuration, which is the first research that constructs the human pose regression model in the framework of supervised tensor learning.
(3) To capture underlying structural information of the visual observation and reduce loss of information, we adopt CP decomposition to this end. This method allows multiple projections of low-dimensional tensor feature to more than one direction along each tensor mode.
(4) To seek for the optimal tensor order automatically during the regression process, we adopt the group-sparisty norm instead of Frobenius-norm in our regression model, which leads to our proposed OOSTR model. Additionally, the group-sparisty norm takes both L1- and L2-norm into consideration simultaneously.

2 Related works

As the rapidly development of multimedia technology, a lot of human pose estimation methods have been presented. In the research of Deutscher et al. [4], they presented a generative

algorithm called annealing particle filter (APF) to infer 3D human pose from the visual observation. It cannot efficiently work because of searching the valid particle sets in the high-dimensional state space for lots of times. Additionally, based on a manifold learning approach presented in [21], it revealed that the locally underlying geometry information is more important than the global structure information since the high-dimensional data is usually embedded in a low-dimensional manifold. The principle component analysis (PCA) is successfully used for dimensionality reduction, and the final pose is estimated by simulated annealing particle swarm optimism (SA-PSO) [19]. But PCA is a linear method of dimensionality reduction that is not very plausible to handle the nonlinear relationship between 2D visual observation and 3D human pose configuration. Yao et al. [35] proposed a Gaussian process latent variable model (GPLVM) to deal with this problem, but GPLVM is not a dynamic model since the temporal structure of the visual observation is not considered. Subsequently, Wang et al. [32] proposed a Gaussian process dynamic model (GPDM) for 3D human motion estimations, but GPDM cannot timely reduce the dimension of a new input and quickly search its corresponding low-dimensional data. Zhao et al. [37] proposed a multi-view visual fusion algorithm and developed an effective CP-SIFT descriptor for 3D human pose regression, which can recover the 3D poses from multi-view image sequence accurately. Nevertheless, the input features used in their research are based on vectors, and thus the underlying structure information of the visual data is drastically destroyed during concatenating the columns and rows of an image matrix. All the aforementioned approaches work in the vector-based space, thus it is likely to suffer from the curse of dimensionality problem. Therefore, it is obvious that these approaches cannot achieve the most significant performance for 3D human pose estimation.

Recently, one important category of methods which has caused widespread attention in machine learning and pattern recognition community is subspace learning, specifically, Hund et al. [8] introduced the subspace clustering to analyze patient groups and immunization results, which is a representative approach and very meaningful for general interactive machine learning analysis. Moreover, Lai et al. [14] extended the subspace learning to a tensor space which considers a given image as a high-order tensor instead of high-dimensional vector. The classical image-to-vector transform process usually results in the curse of dimensionality problems, and the underlying structure information embedded in an image is seriously destroyed after concatenating the columns and rows of an image matrix into a high-dimensional vector [36]. Nevertheless, the underlying structure information plays a central role in determining 3D spatial location information of body joint centers and is beneficial to improve the performance of tracking system. Tensor data usually contains a large amount of redundant information or noise disturbance and not all information or features are necessary for our work at hand. Therefore, a method that can well reduce the redundant information or filter out random noise features should be presented in order to recover human poses effectively and accurately. It is not hard to see that sparse representation (SR) method has been widely used for feature extraction in computer vision community [34]. Through introducing L1-norm into the objection function of SR, most of sparse coefficients are shrunk to be zero. Thus, the objective of sparse feature selection or dimensionality reduction is achieved. Lai et al. [12] presented a novel algorithm called sparse alignment for robust tensor learning (STA). They introduced the sparse alignment techniques to unify several tensor learning models, and finally proposed a novel tensor learning model for face recognition.

But sparsity has not been encoded to the tensor subspaces. Lai et al. [13] presented a sparse discriminant projection learning (SDPL) model for human gait recognition, which introduced the sparse representation into tensor subspace learning framework. However, until now, designing a robust sparse tensor subspace learning algorithm to reconstruct the 3D human poses has not been done. Inspired by the existing works, we propose a novel discriminative model that combines RSTSL with OOSTR to accurately estimate 3D human pose configurations from multi-view image or video sequence.

3 Notations and preliminaries

Some typical definitions, notions and concepts of tensor algebra that are very similar to those presented in [11] will be described. Throughout this paper, lowercase or uppercase italic letters represent scalars (e.g., a, B), boldface lowercase letters represent vectors (e.g., a, b) and boldface uppercase letters represent matrices (e.g., A, B). Tensors are able to be considered as multi-dimensional arrays and will be represented by calligraphy uppercase letters (e. g., A, B).

Assume that the training samples are represented as a set of N-order tensors $\left\{\mathrm{A}_i \in \mathbb{R}^{h_1 \times h_2 \times \cdots \times h_N}, i = 1, \ldots, n\right\}$, where n is the total number of training samples. We provide four definitions that are close to [11] for discussions below.

Definition 1 The inner product of two equivalent tensors $\mathrm{A}, \mathrm{B} \in \mathbb{R}^{h_1 \times h_2 \times \cdots \times h_N}$ is denoted as <A, B>, and then we can have $< \mathrm{A}, \mathrm{B} >= \sum_{i_1=\cdots=i_N=1}^{h_1 \times h_2 \times \cdots \times h_N} \mathrm{A}_{i_1,\ldots,i_N} \mathrm{B}_{i_1,\ldots,i_N}$. The Frobenius - norm of a tensor A is denoted as a square root of inner product, i.e. $||\mathrm{A}||_{\mathrm{F}} = \sqrt{< \mathrm{A}, \mathrm{A} >}$. For any mode - k of a tensor A, we have $||\mathrm{A}||_{\mathrm{F}} = ||\mathbf{A}^{(k)}||_{\mathrm{F}} = \sqrt{\mathbf{A}^{(k)}\left(\mathbf{A}^{(k)}\right)^T}$.

Definition 2 The any mode - k unfolding of a tensor A into the matrix $\mathbf{A}^{(k)} \in \mathbb{R}^{h_k \times \prod_{i \neq k} h_i}$, i.e. $\mathbf{A}^{(k)} \Leftarrow_k \mathrm{A}$, can be represented as $\mathbf{A}^{(k)}_{i_k, j} = \mathrm{A}_{i_1 \ldots i_N}$, where $j = 1 + \sum_{l=1, l \neq k}^{N} (i_l - 1) \prod_{Q=l+1, Q \neq k}^{l-1} h_Q$. The schematic description is illustrated in Fig. 1 for a 3-order tensor A.

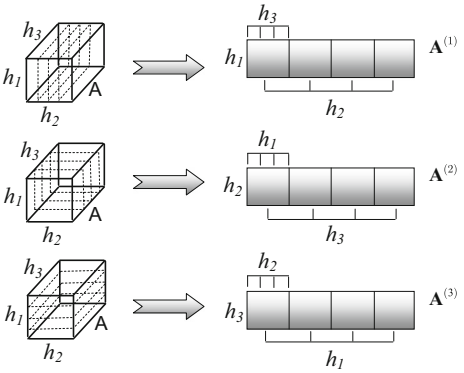

Fig. 1 Unfolding of a tensor $\mathrm{A} \in \mathbb{R}^{h_1 \times h_2 \times h_3}$ into the matrix $\mathbf{A}^{(1)} \in \mathbb{R}^{h_1 \times h_2 h_3}$, $\mathbf{A}^{(2)} \in \mathbb{R}^{h_2 \times h_3 h_1}$ and $\mathbf{A}^{(3)} \in \mathbb{R}^{h_3 \times h_1 h_2}$

Definition 3 The mode-k product of a N-order tensor A with the matrix with the matrix $\mathbf{M}\in\mathbb{R}^{h'_k\times h_k}$ is represented as $\mathrm{B}=\mathrm{A}\times_k\mathbf{M}$, i.e. $\mathrm{B}_{i_1,\ldots,i_{k-1},i,i_{k+1},\ldots,i_N}=\sum_{j=1}^{h'_k}\mathrm{A}_{i_1,\ldots,i_{k-1},j,i_{k+1},\ldots,i_N}\mathbf{M}_{i,j},\ j=1,2,\ldots,h'_k$, where $\mathbf{M}_{ij}$ denotes the element in the matrix $\mathbf{M}$ of coordinate (i,j). The unfolded form of this product is represented as $\mathrm{B}=\mathrm{A}\times_k\mathbf{M}\Leftrightarrow\mathbf{U}\mathbf{A}^{(k)}$. The multiplication in every mode except mode-k is formulated as

$$\mathrm{A}\times_1\mathbf{M}_1\times\cdots\times_{k-1}\mathbf{M}_{k-1}\times_{k+1}\mathbf{M}_{k+1}\times\cdots\times_N\mathbf{M}_N\triangleq\mathrm{A}\prod_{d=1,d\neq k}^{N}\times_d\mathbf{M}_d\triangleq\mathrm{A}\overline{\times}_k\mathbf{M}_k \quad (1)$$

The tensor B can be decomposed as $\mathrm{B}=\mathrm{A}\times_1\mathbf{M}_1\times_2\mathbf{M}_2\times\cdots\times_N\mathbf{M}_N$ in every mode, where $\{\mathbf{M}_l\}_{l=1}^N$ denotes a set of orthogonal matrices that involve the ordered principal components and the tensor A is a core tensor. Operator "$\otimes$" denotes Kronecker-product. The mode - k unfolding of a tensor B is represented as

$$\mathbf{B}^{(k)}=\mathbf{M}_k\mathbf{A}^{(k)}\left(\mathbf{M}_N\otimes\cdots\otimes\mathbf{M}_{k+1}\otimes\mathbf{M}_{k-1}\otimes\cdots\otimes\mathbf{M}_1\right)^T \quad (2)$$

Definition 4 The CP (Canonical/Parallel-Factor) decomposition factories a N-order tensor $\mathrm{A}\in\mathbb{R}^{h_1\times h_2\times\cdots\times h_N}$ into a linear combination of Z rank-1 tensors, denoted as

$$\mathrm{A}\approx\sum_{z=1}^{Z}\mathbf{m}_z^{(1)}\circ\mathbf{m}_z^{(2)}\circ\cdots\circ\mathbf{m}_z^{(N)}\triangleq\left(\mathbf{M}^{(1)},\mathbf{M}^{(2)},\ldots,\mathbf{M}^{(N)}\right) \quad (3)$$

Operator "$\circ$" is the outer product of vectors and the matrix $\mathbf{M}^{(k)}=\left[\mathbf{m}_1^{(k)},\mathbf{m}_2^{(k)},\ldots,\mathbf{m}_Z^{(k)}\right]\in\mathbb{R}^{h_k\times Z},k=1,2,\ldots,N$, the CP decomposition of mode - k unfolded tensor can be formulated as

$$\mathbf{A}^{(k)}=\mathbf{M}^{(k)}\left(\mathbf{M}^{(N)}\odot\cdots\odot\mathbf{M}^{(k+1)}\odot\mathbf{M}^{(k-1)}\odot\cdots\odot\mathbf{M}^{(1)}\right)^T \quad (4)$$

Where operator "$\odot$" denotes Khatri-Rao product, the rank of a tensor A is represented as rank (A). The schematic description of CP decomposition is illustrated in Fig. 2 for a 3-order tensor (i.e., $N=3$), which is corresponding to (3). The CP decomposition is a linear combination of Z rank −1 tensors, e.g., z-th rank -1 tensor $\{\mathbf{m}_z^{(k)}\}_{k=1}^{N=3}$.

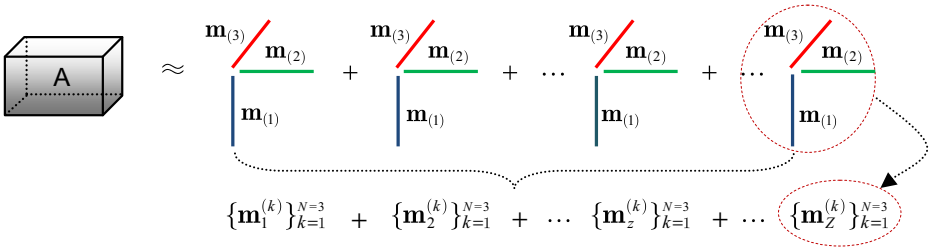

Fig. 2 Schematic description of CP decomposition

4 Human body models and visual observations

4.1 3D human body models

In order to promote the subsequent discussions, we briefly describe the corresponding 3D human body models that are applied to simulate human motions in 3D. These models are based on the kinematic chain as shown in Fig. 3a. In this paper, we adopt two types of articulated human models: (1) the first one from Brown database [26] is composed of 10 rigid body limbs and every two limbs are connected by a joint as illustrated in Fig. 3b, including head, torso, upper and lower arms (right and left), thighs and calves (right and left); (2) the second one from HumanEva [27] is composed of 15 rigid body limbs as shown in Fig. 3c, including head, torso, pelvis, upper and lower arms (right and left), thighs and calves (right and left), hands and feet (right and left). The each body limb is modeled by a cylinder.

4.2 Representation of visual observations

According to the recently proposed works [12–14, 34, 36], we can find that a tensor can be treated as a natural representation of the visual observation. A color (or gray) image can be regarded as a 3-order (or 2-order) tensor, the mode-1 and mode-2 of a tensor denote the height and the width of an image, respectively. For a color image, the mode-3 of a tensor denotes the color space [29]. In this paper, the final pose is estimated by a robust regression algorithm which highly depends on the representation of the visual observation. In addition, HOGs (Histogram of Oriented Gradients) has been widely used in many regression problems and achieved a good performance [17]. Naturally, we also introduce the HOG descriptor into the representation of an image in this paper. Some illustrations are shown in Fig. 4.

5 The overview of our tracking system

We provide the framework of our tracking system as shown in Fig. 5, the framework consists of two major components, i.e., training and testing. Our tracking system is aimed at extracting the most important low-dimensional tensor feature from an input

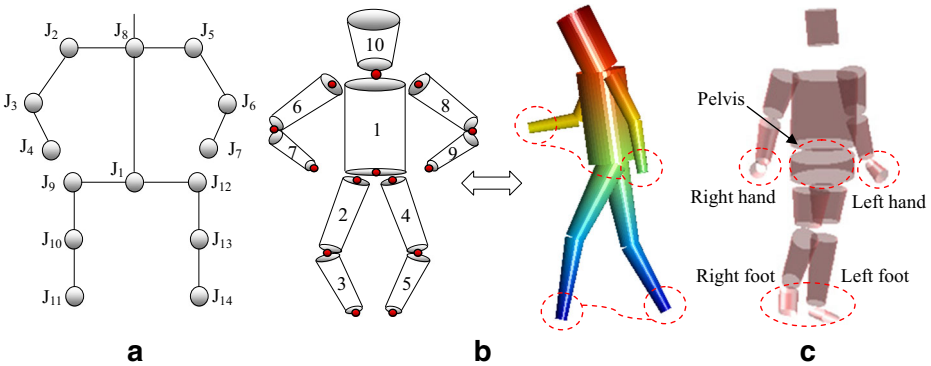

Fig. 3 3D human body models

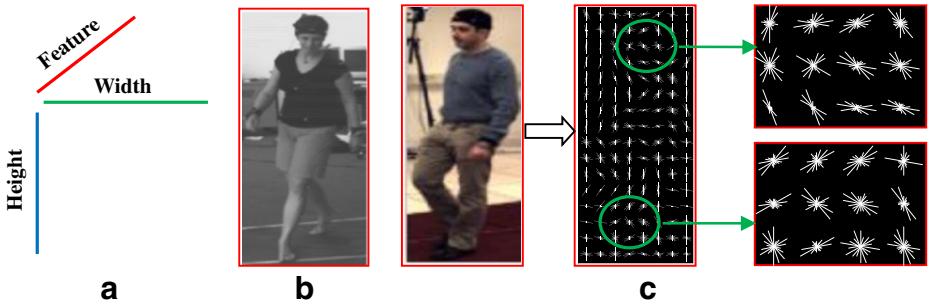

Fig. 4 Illustrations of the visual observation represented as tensors. (**a**) 3-mode tensor-based representation. (**b**) A gray image of a human subject (i.e. 2-order tensor). (**c**) A color image of a human subject and the corresponding HOGs (i.e. 3-order tensor)

image using our proposed RSTSL and constructing OOSTR regression model that is used to build a finest mapping between the extracted low-dimensional tensor feature and the human pose configuration. Our OOSTR model is essentially a local linear model. Therefore, it is not directly compatible with global nonlinear relationship between the visual feature and the human pose configuration. In order to handle this problem, a finite number of local linear models are trained by the affinity-propagation method (AP) [5]. Initially, the training samples are clustered in pose state space, and then the samples with close state configuration are gathered in the same clusters. Finally, we introduce the random forest [16] as the multi-class classifier of 3D human pose classes defined by the corresponding training sample clusters. This classifier can efficiently handle high-dimensional data, which is beneficial to train those local linear models.

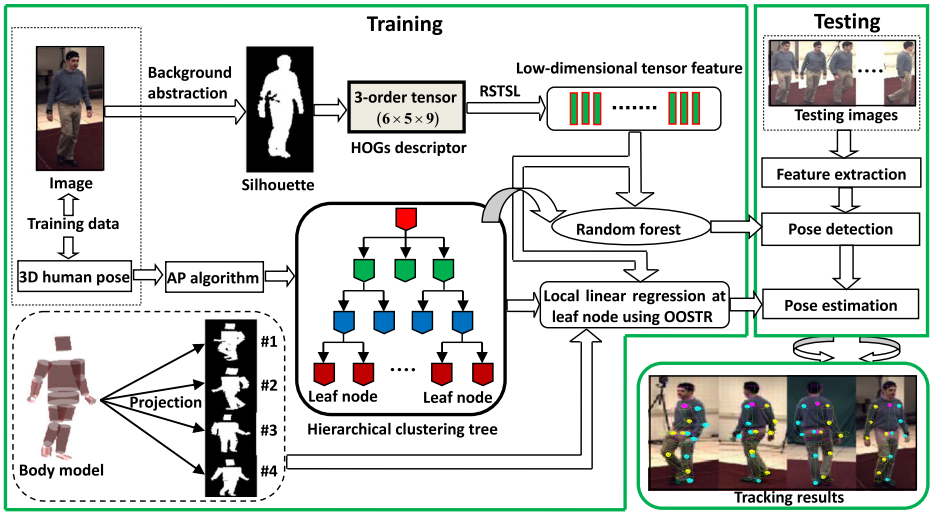

Fig. 5 The overview of our tracking system

6 Robust sparse tensor subspace learning (RSTSL)

6.1 Image representation and parameterization of 3D human pose

In this paper, the visual feature is based on HOGs. To extract HOGs, the silhouette of a human subject is first extracted from an input image through background subtraction. Then, the bounding box containing the human body silhouette is divided into 6×5 sub-images and the gradient orientations of each sub-image are subdivided into 9 bins. Finally, each input image is defined as a $6 \times 5 \times 9$ tensor A of order-3. The 3D human pose used for Brown database is modeled as a 42-dimensional vector y that is assigned to 14 joint centers where each joint has 3 DOF (Degree of Freedom). The 3D human pose used for HumanEva database is modeled by 34 parameters which consist of two modules, the one involves global translation and rotation of the pelvis and another one involves relative joint angles between adjacent joints.

6.2 Low-dimensional tensor feature extraction

Let P is the total number of classes of human motions, e.g. *Walking*, *Jogging*, *Boxing*, *Gestures* and *Throw/Catch* and so on. A_i^p is an image of class p, Q_p is the number of images of class p. In this paper, our main task is to seek N projection matrices $\{\mathbf{M}_l\}_{l=1}^N$ to complete the projection transformation below:

$$\mathrm{B}_i^p = \mathrm{A}_i^p \times_1 \mathbf{M}_1 \times_2 \mathbf{M}_2 \times \cdots \times_N \mathbf{M}_N, \;\; i = 1, \ldots, Q_p, \;\; p = 1, \ldots, P \tag{5}$$

Let Q is the total number of images of all classes. The Eq. (5) ensures that the projected tensors from the same classes are distributed as close as possible while the ones from the different classes are distributed as far as possible. Suppose that $\overline{\mathrm{A}}_p$ is the mean value of images of class p and $\overline{\mathrm{A}}$ is the mean value of images of all classes. Thus, the between-class divergence $\mathbf{\Omega}_b(\mathrm{A})$ and within-class divergence $\mathbf{\Omega}_w(\mathrm{A})$ of an image are formulated as

$$\begin{cases} \mathbf{\Omega}_b(\mathrm{A}) = \sum\limits_p Q_p \left\| \overline{\mathrm{A}}_p - \overline{\mathrm{A}} \right\|_{\mathrm{F}}^2 \\ \mathbf{\Omega}_w(\mathrm{A}) = \sum\limits_i \left\| \mathrm{A}_i^p - \overline{\mathrm{A}}_{p_i} \right\|_{\mathrm{F}}^2 \end{cases} \tag{6}$$

A natural idea is to minimize within-class divergence $\mathbf{\Omega}_w(\mathrm{B})$ of the projected tensor B whilst maximize between-class divergence $\mathbf{\Omega}_b(\mathrm{B})$, thus we can construct the objective function, denoted as

$$\left(\mathbf{M}_k^* \big|_{k=1}^N\right) = \arg \max_{\mathbf{M}_k \big|_{k=1}^N} \frac{\mathbf{\Omega}_b(\mathrm{B})}{\mathbf{\Omega}_w(\mathrm{B})} \tag{7}$$

The numerator and denominator of (7) are reformulated to (8), respectively

$$\begin{cases} \mathbf{\Omega}_b(\mathrm{B}) = \sum_p Q_p \left\| \overline{\mathrm{A}}_p \times_1 \mathbf{M}_1 \times \cdots \times_N \mathbf{M}_N - \overline{\mathrm{A}} \times_1 \mathbf{M}_1 \times \cdots \times_N \mathbf{M}_N \right\|_{\mathrm{F}}^2 \\ \mathbf{\Omega}_w(\mathrm{B}) = \sum_i \left\| \mathrm{A}_i^p \times_1 \mathbf{M}_1 \times \cdots \times_N \mathbf{M}_N - \overline{\mathrm{A}}_{p_i} \times_1 \mathbf{M}_1 \times \cdots \times_N \mathbf{M}_N \right\|_{\mathrm{F}}^2 \end{cases} \tag{8}$$

Now, we discuss how to optimize (7) along mode - k. Combining Eq. (5) with *Definition 2*, (8) is rewritten as

$$\begin{cases} \mathbf{\Omega}_b^{(k)}(\mathrm{B}) = \sum_p Q_p \left\| \overline{\mathrm{A}}_p \times_k \mathbf{M}_k - \overline{\mathrm{A}} \times_k \mathbf{M}_k \right\|_{\mathrm{F}}^2 \\ \mathbf{\Omega}_w^{(k)}(\mathrm{B}) = \sum_i \left\| \mathrm{A}_i^p \times_k \mathbf{M}_k - \overline{\mathrm{A}}_{p_i} \times_k \mathbf{M}_k \right\|_{\mathrm{F}}^2 \end{cases} \tag{9}$$

Theorem 1 For an arbitrary square matrix A, we have $\|\mathbf{A}\|_{\mathrm{F}}^2 = \mathrm{Tr}(\mathbf{A}\mathbf{A}^T)$, thus, (7) can be rewritten as

$$\left(\mathbf{M}_k^*\right) = \arg\max_{\mathbf{M}_k} \frac{\mathrm{Tr}\left(\mathbf{M}_k^T \mathbf{G}_b^{(k)} \mathbf{M}_k\right)}{\mathrm{Tr}\left(\mathbf{M}_k^T \mathbf{G}_w^{(k)} \mathbf{M}_k\right)} \tag{10}$$

We define the mode - k between-class divergence matrix $\mathbf{G}_b^{(k)}$ and within-class divergence matrix $\mathbf{G}_w^{(k)}$ as

$$\begin{cases} \mathbf{G}_b^{(k)} = \sum_{j=1}^{\prod_{t\neq k} h_t} \mathbf{G}_b^{j,(k)}, \ \mathbf{G}_b^{j,(k)} = \sum_{p=1}^{n_p} Q_p \left(\overline{\mathbf{A}}_p^{j,(k)} - \overline{\mathbf{A}}^{j,(k)} \right) \mathbf{M}_k \mathbf{M}_k^T \left(\overline{\mathbf{A}}_p^{j,(k)} - \overline{\mathbf{A}}^{j,(k)} \right)^T \\ \mathbf{G}_w^{(k)} = \sum_{j=1}^{\prod_{t\neq k} h_t} \mathbf{G}_w^{j,(k)}, \ \mathbf{G}_w^{j,(k)} = \sum_{i=1}^{n} \left(\mathbf{A}_{i,p}^{j,(k)} - \overline{\mathbf{A}}_{p_i}^{j,(k)} \right) \mathbf{M}_k \mathbf{M}_k^T \left(\mathbf{A}_{i,p}^{j,(k)} - \overline{\mathbf{A}}_{p_i}^{j,(k)} \right)^T \end{cases} \tag{11}$$

The tensor A belongs to the class indexed as $p_i = [1,2,\ldots,n_p]$, where n_p is the total number of images of class P. And $\mathbf{M}_k = \mathbf{M}_N \otimes \cdots \otimes \mathbf{M}_{k+1} \otimes \mathbf{M}_{k-1} \otimes \cdots \otimes \mathbf{M}_1$, $k = 1, 2, \ldots, N$.

Proof Through a simple tensor algebra operation, we can have $||\mathrm{A}\times_k\mathbf{M}||_\mathrm{F} = ||\left(\mathbf{A}^{(k)}\right)^T\mathbf{M}||_\mathrm{F}$, where $\mathbf{A}^{(k)}$ denotes the mode - k unfolded matrix of a tensor A. Thus the mode - k within-class divergence matrix in (9) is reformulated to

$$\begin{aligned}
\mathbf{\Omega}_w^{(k)}(\mathrm{B}) &= \sum_i \left\|\mathrm{A}_i^p\times_k\mathbf{M}_k - \overline{\mathrm{A}}_{p_i}\times_k\mathbf{M}_k\right\|_\mathrm{F}^2 \\
&= \sum_i \left\|\mathbf{M}_k^T\left(\mathbf{A}_{i,p}^{(k)} - \overline{\mathbf{A}}_{p_i}^{(k)}\right)\mathbf{M}_k\right\|_\mathrm{F}^2 \\
&= \sum_i \mathrm{Tr}\left[\mathbf{M}_k^T\left(\left(\mathbf{A}_{i,p}^{(k)} - \overline{\mathbf{A}}_{p_i}^{(k)}\right)\mathbf{M}_k\mathbf{M}_k^T\left(\mathbf{A}_{i,p}^{(k)} - \overline{\mathbf{A}}_{p_i}^{(k)}\right)^T\right)\mathbf{M}_k\right] \\
&= \mathrm{Tr}\left[\mathbf{M}_k^T\left(\sum_i\left(\mathbf{A}_{i,p}^{(k)} - \overline{\mathbf{A}}_{p_i}^{(k)}\right)\mathbf{M}_k\mathbf{M}_k^T\left(\mathbf{A}_{i,p}^{(k)} - \overline{\mathbf{A}}_{p_i}^{(k)}\right)^T\right)\mathbf{M}_k\right] \\
&= \mathrm{Tr}\left[\mathbf{M}_k^T\left(\sum_{j=1}^{\prod_{t\neq k}h_t}\sum_i\left(\mathbf{A}_{i,p}^{j,(k)} - \overline{\mathbf{A}}_{p_i}^{j,(k)}\right)\mathbf{M}_k\mathbf{M}_k^T\left(\mathbf{A}_{i,p}^{j,(k)} - \overline{\mathbf{A}}_{p_i}^{j,(k)}\right)^T\right)\mathbf{M}_k\right] \\
&= \mathrm{Tr}\left(\mathbf{M}_k^T\mathbf{G}_w^{(k)}\mathbf{M}_k\right)
\end{aligned} \tag{12}$$

In the same way, we can also prove $\mathbf{\Omega}_b^{(k)}(\mathrm{B}) = \sum_p Q_p||\overline{\mathrm{A}}_p\times_k\mathbf{M}_k - \overline{\mathrm{A}}\times_k\mathbf{M}_k||_\mathrm{F}^2 = \mathrm{Tr}\left(\mathbf{M}_k^T\mathbf{G}_b^{(k)}\mathbf{M}_k\right)$.

Refer to the theory of Rayleigh-quotient [7]. The Eq. (10) can be optimized only when the matrix M_k is composed of l_k eigenvectors that are associated with l_k largest eigenvalues of the matrix pairs $(\mathbf{G}_b^{(k)}, \mathbf{G}_w^{(k)})$. The l_k eigenvectors can be obtained by resolving the following problem:

$$\mathbf{G}_b^{(k)}\cdot\boldsymbol{\xi} = \alpha\,\mathbf{G}_w^{(k)}\cdot\boldsymbol{\xi} \tag{13}$$

Nevertheless, the solutions of (7) are not sparse. We not only calculate a set of matrices $\{\mathbf{M}_k\}_{l=1}^N$ but also ensure that they are sparse, i.e., the most of elements of M_k are zero. Thus we need to impose a sparse constraint on (10), then

$$\begin{cases} \left(\mathbf{M}_k^*|_{k=1}^N\right) = \arg\max\limits_{\mathbf{M}_k|_{k=1}^N} \dfrac{\mathrm{Tr}\left(\mathbf{M}_k^T\mathbf{G}_b^{(k)}\mathbf{M}_k\right)}{\mathrm{Tr}\left(\mathbf{M}_k^T\mathbf{G}_w^{(k)}\mathbf{M}_k\right)} \\ s.t.\ \ \mathrm{R}(\mathbf{M}_k) < \mathrm{K}_{(k)}\ ,\ k = 1,\ldots,N \end{cases} \tag{14}$$

Where function $R(M_k)$ is used to count the number of non-zero elements of the matrix M_k. In addition, we provide two $(N+1)$- order tensor $D_w \in \mathbb{R}^{h_1 \times \cdots \times h_N \times Q}$ and $D_b \in \mathbb{R}^{h_1 \times \cdots \times h_N \times Q}$, denoted as

$$D_w = \left\{ \begin{array}{c} \left(A_1^1 - \overline{A}_1\right) \\ \vdots \\ \left(A_{Q_1}^1 - \overline{A}_1\right) \\ \vdots \\ \left(A_1^P - \overline{A}_{n_P}\right) \\ \vdots \\ \left(A_{Q_P}^P - \overline{A}_{n_P}\right) \end{array} \right\} \quad \text{and} \quad D_b = \left\{ \begin{array}{c} \left(\overline{A}_1 - \overline{A}\right) \\ \vdots \\ \left(\overline{A}_1 - \overline{A}\right) \\ \vdots \\ \left(\overline{A}_P - \overline{A}\right) \\ \vdots \\ \left(\overline{A}_P - \overline{A}\right) \end{array} \right\} \tag{15}$$

Theorem 2 Given $N-1$ sparse projection matrices $\mathbf{M}_1, \ldots, \mathbf{M}_{k-1}, \mathbf{M}_{k+1}, \ldots, \mathbf{M}_N$, then D_w and D_b can be projected to low-dimensional tensor subspace through sparse projection transformation, denoted as

$$\begin{cases} W_w = D_w \times_1 \mathbf{M}_1 \times \cdots \times_{k-1} \mathbf{M}_{k-1} \times_{k+1} \mathbf{M}_{k+1} \times \cdots \times_N \mathbf{M}_N \\ W_b = D_b \times_1 \mathbf{M}_1 \times \cdots \times_{k-1} \mathbf{M}_{k-1} \times_{k+1} \mathbf{M}_{k+1} \times \cdots \times_N \mathbf{M}_N \end{cases} \tag{16}$$

Finally, we can have $\mathbf{G}_w^{(k)} = \mathbf{W}_w^{(k)} \cdot (\mathbf{W}_w^{(k)})^T$ and $\mathbf{G}_b^{(k)} = \mathbf{W}_b^{(k)} \cdot (\mathbf{W}_b^{(k)})^T$.

Proof The mode - k unfolding of the tensor D_w into the matrix $\mathbf{W}_w^{(k)}$ is denoted as

$$\mathbf{W}_w^{(k)} = \left(D_w \times_1 \mathbf{M}_1 \times \cdots \times_{k-1} \mathbf{M}_{k-1} \times_{k+1} \mathbf{M}_{k+1} \times \cdots \times_N \mathbf{M}_N\right)^{(k)} \tag{17}$$

For the human motion of class p, we can obtain

$$\begin{aligned} &\left(\sum_i \left\| A_i^p - \overline{A}_{p_i} \right\|_F^2 \times_1 \mathbf{M}_1 \times \cdots \times_{k-1} \mathbf{M}_{k-1} \times_{k+1} \mathbf{M}_{k+1} \times \cdots \times_N \mathbf{M}_N \right)^{(k)} \\ &= \left(\sum_i \left\| A_i^p - \overline{A}_{p_i} \right\|_F^2 \right)^{(k)} \cdot \mathbf{M}_k \\ &= \left(\sum_i \left\| \mathbf{A}_{i,p}^{(k)} - \overline{\mathbf{A}}_{p_i}^{(k)} \right\|_F^2 \right) \cdot \mathbf{M}_k \end{aligned} \tag{18}$$

Therefore, Eq. (17) can be reformulated to

$$\mathbf{W}_w^{(k)} = \left[\left(\sum_i \left\| \mathbf{A}_{i,p}^{1,(k)} - \overline{\mathbf{A}}_{p_i}^{1,(k)} \right\|_F^2 \right) \cdot \mathbf{M}_k, \ldots, \left(\sum_i \left\| \mathbf{A}_{i,p}^{\prod_{t \neq k} h_{t,(k)}} - \overline{\mathbf{A}}_{p_i}^{\prod_{t \neq k} h_{t,(k)}} \right\|_F^2 \right) \cdot \mathbf{M}_k \right] \tag{19}$$

The product of the matrix $\mathbf{W}_w^{(k)}$ with $(\mathbf{W}_w^{(k)})^T$ is represented as

$$\begin{aligned}
&\mathbf{W}_w^{(k)}\cdot\left(\mathbf{W}_w^{(k)}\right)^T = \\
&\left[\left(\sum_i\left\|\mathbf{A}_{i,p}^{1,(k)}-\overline{\mathbf{A}}_{p_i}^{1,(k)}\right\|_F^2\right)\cdot\mathbf{M}_k,\ldots,\left(\sum_i\left\|\mathbf{A}_{i,p}^{\prod_{t\neq k}h_t,(k)}-\overline{\mathbf{A}}_{p_i}^{\prod_{t\neq k}h_t,(k)}\right\|_F^2\right)\cdot\mathbf{M}_k\right]\times \\
&\left[\begin{array}{c}\mathbf{M}_k^T\cdot\left(\sum_i\left\|\mathbf{A}_{i,p}^{1,(k)}-\overline{\mathbf{A}}_{p_i}^{1,(k)}\right\|_F^2\right)^T\\ \vdots\\ \mathbf{M}_k^T\cdot\left(\sum_i\left\|\mathbf{A}_{i,p}^{\prod_{t\neq k}h_t,(k)}-\overline{\mathbf{A}}_{p_i}^{\prod_{t\neq k}h_t,(k)}\right\|_F^2\right)^T\end{array}\right] = \sum_{j=1}^{\prod_{t\neq k}h_t}\sum_i\left(\mathbf{A}_{i,p}^{j,(k)}-\overline{\mathbf{A}}_{p_i}^{j,(k)}\right)\mathbf{M}_k\mathbf{M}_k^T\left(\mathbf{A}_{i,p}^{j,(k)}-\overline{\mathbf{A}}_{p_i}^{j,(k)}\right)^T \\
&\qquad\qquad = \mathbf{G}_w^{(k)}
\end{aligned} \tag{20}$$

Therefore, we can obtain $\mathbf{G}_w^{(k)}=\mathbf{W}_w^{(k)}\cdot(\mathbf{W}_w^{(k)})^T$. Similarly, we can also prove $\mathbf{G}_b^{(k)}=\mathbf{W}_b^{(k)}\cdot(\mathbf{W}_b^{(k)})^T$.

Lemma 1 Let Y be a $h_k\times h_k$ symmetric positive semi-definite matrix and $h_k > l_k$. The eigenvalues of the matrix Y satisfy $z_{11}\geq\cdots\geq z_{l_kl_k} > z_{(l_k+1)(l_k+1)}\geq\cdots\geq z_{h_kh_k}\geq 0$, therefore, Tr (X^TYX) is able to be maximized at X subject to the constraint $X^TX=1$, where X is a $h_k\times l_k$ matrix which is defined as $X=V_1U_1$, and V_1 is composed of leading l_k eigenvectors of the matrix Y and U_1 is an arbitrary $l_k\times l_k$ orthogonal matrix.

Proof Suppose that the eigenvalue decomposition of Y is denoted as $Y=VZV^T$, where $\mathbf{V}\in\mathbb{R}^{h_k\times h_k}$ is orthogonal and $Z=\mathrm{diag}\ (z_{11},\ldots,z_{h_kh_k})$ is diagonal, respectively. Let $\mathbf{U}=\mathbf{V}^T\mathbf{X}\in\mathbb{R}^{h_k\times l_k}$, we can have Tr $(X^TYX)=$ Tr (X^TVZV^TX) and $X=VU$. Through a simple matrix operation, we can also obtain $U^TU=(V^TX)^T(V^TX)=X^TVV^TX=I$, thus we can find that U has the orthogonal columns. Additionally, we denote the rows of the matrix U as $\mathbf{u}_1^T,\mathbf{u}_2^T,\ldots,\mathbf{u}_{hk}^T$, then

$$\mathrm{Tr}\left(\mathbf{X}^T\mathbf{Y}\mathbf{X}\right)=\mathrm{Tr}\left(\mathbf{X}^T\mathbf{V}\mathbf{Z}\mathbf{V}^T\mathbf{X}\right)=\mathrm{Tr}\left(\mathbf{U}^T\mathbf{Z}\mathbf{U}\right)=\sum_{i=1}^{h_k}z_{ii}\mathrm{Tr}\left(\mathbf{u}_i\mathbf{u}_i^T\right)=\sum_{i=1}^{h_k}z_{ii}|\mathbf{u}_i|^2 \tag{21}$$

Since the orthogonal columns of the matrix U satisfy $|\mathbf{u}_i|^2\leq 1,\ i=1,\ldots,h_k$ and $\sum_{i=1}^{h_k}|\mathbf{u}_i|^2=l_k$. Therefore, the Eq. (21) is simplified to a maximization problem subject to the constraint $|\mathbf{u}_i|^2\leq 1,\ i=1,\ldots,h_k$ and $\sum_{i=1}^{h_k}|\mathbf{u}_i|^2=l_k$, which can be resolved by $|\mathbf{u}_1|=\cdots=|\mathbf{u}_{l_k}|=1$ and $|\mathbf{u}_{l_k+1}|=\cdots=|\mathbf{u}_{h_k}|=0$. The first l_k rows form the matrix $\mathbf{U}_1\in\mathbb{R}^{l_k\times l_k}$ while the rest rows of the matrix U are zero. Let $\mathbf{V}=(\mathbf{V}_1,\mathbf{V}_2)$, where $\mathbf{V}_1\in\mathbb{R}^{h_k\times l_k}$. Finally, we can obtain $X=V_1U_1$.

Theorem 3 Let $\mathbf{G}_w^{(k)}$ be a positive semi-definite matrix, the Cholesky decomposition [7] of matrix $\mathbf{G}_w^{(k)}$ is defined as $\mathbf{G}_w^{(k)}=\mathbf{S}_{w(k)}^T\mathbf{S}_{w(k)}$, where $\mathbf{S}_{w(k)}\in\mathbb{R}^{h_k\times h_k}$ is an upper - triangular matrix. Let $\xi_1,\ldots,\xi_{l_k}$ be the eigenvectors of Eq. (13), which are corresponding to the l_k largest

eigenvalues. Let $\mathbf{E}_{h_k \times l_k} = \left[\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{l_k}\right]$ and $F_{h_k \times l_k} = [v_1, \ldots, v_{l_k}]$. For $\lambda > 0$, and $\mathbf{F}$ are the optimal solutions of (22)

$$\min_{\mathbf{E},\mathbf{F}} \left(\sum_{i=1}^{h_k} \left\| \mathbf{S}_{w(k)}^{-T} \mathbf{W}_b^{(k)}(i,:) - \mathbf{E}\mathbf{F}^T \mathbf{W}_b^{(k)}(i,:) \right\|^2 + \lambda \sum_{j=1}^{l_k} \boldsymbol{\nu}_j^T \mathbf{G}_w^{(k)} \boldsymbol{\nu}_j \right), \quad s.t. \ \mathbf{E}^T\mathbf{E} = \mathbf{I} \tag{22}$$

Where $h_k = h_1 \times \cdots \times h_{k-1} \times h_{k+1} \times \cdots \times h_N \times Q$, and $\boldsymbol{\nu}_j$, $j = 1, \ldots, l_k$ spans the same subspace as ξ_j, $j = 1, \ldots, l_k$. If $\mathbf{E}$ is fixed, the optimal solution of (22) is written as $\mathbf{F} = \left(\mathbf{G}_b^{(k)} + \lambda \mathbf{G}_w^{(k)}\right)^{-1} \mathbf{G}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \mathbf{E}$.

Proof We substitute $\mathbf{F}$ into (22) and then obtain the objective function we want to optimize

$$\mathrm{Tr}\left\{ \mathbf{E}^T \mathbf{S}_{w(k)}^{-T} \mathbf{G}_b^{(k)} \left(\mathbf{G}_b^{(k)} + \lambda \mathbf{G}_w^{(k)}\right)^{-1} \mathbf{G}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \mathbf{E} \right\}, \quad s.t. \ \mathbf{E}^T\mathbf{E} = \mathbf{I} \tag{23}$$

The leading l_k eigenvectors of $\mathbf{S}_{w(k)}^{-T} \mathbf{G}_b^{(k)} \mathbf{S}_{w(k)}^{-1}$ are concatenated into a matrix $\boldsymbol{\Sigma} = [\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \ldots, \boldsymbol{\sigma}_{l_k}]$ and then we can obtain $\mathbf{S}_{w(k)}^{-T} \mathbf{G}_b^{(k)} \mathbf{S}_{w(k)}^{-1} = \boldsymbol{\Sigma}\boldsymbol{\Lambda}\boldsymbol{\Sigma}^T$, where $\boldsymbol{\Lambda} \in \mathbb{R}^{l_k \times l_k}$ is a diagonal matrix of eigenvalues. Thus the leading l_k eigenvectors of (24) are corresponding to the columns of the matrix $\boldsymbol{\Sigma}$

$$\begin{aligned} &\mathbf{S}_{w(k)}^{-T} \mathbf{G}_b^{(k)} \left(\mathbf{G}_b^{(k)} + \lambda \mathbf{G}_w^{(k)}\right)^{-1} \mathbf{G}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \\ &= \mathbf{S}_{w(k)}^{-T} \mathbf{G}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \left(\mathbf{S}_{w(k)}^{-T} \mathbf{G}_b^{(k)} \mathbf{S}_{w(k)}^{-1} + \lambda \mathbf{I}\right)^{-1} \mathbf{S}_{w(k)}^{-T} \mathbf{G}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \end{aligned} \tag{24}$$

According to *Lemmal*, that can maximize (23) satisfies $= \boldsymbol{\Sigma}\mathbf{P}$, where $\mathbf{P}$ is an arbitrary $l_k \times l_k$ orthogonal matrix, substituting into $\mathbf{F}$, we can have

$$\begin{aligned} \mathbf{F} &= \mathbf{S}_{w(k)}^{-1} \left(\mathbf{S}_{w(k)}^{-T} \mathbf{G}_b^{(k)} \mathbf{S}_{w(k)}^{-1} + \lambda \mathbf{I}\right)^{-1} \mathbf{S}_{w(k)}^{-T} \mathbf{G}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \mathbf{E} \\ &= \mathbf{S}_{w(k)}^{-1} \left(\boldsymbol{\Sigma}\boldsymbol{\Lambda}\boldsymbol{\Sigma}^T + \lambda \mathbf{I}\right)^{-1} \boldsymbol{\Sigma}\boldsymbol{\Lambda}\boldsymbol{\Sigma}^T \boldsymbol{\Sigma}\mathbf{P} \\ &= \mathbf{S}_{w(k)}^{-1} \boldsymbol{\Sigma} (\boldsymbol{\Lambda} + \lambda \mathbf{I})^{-1} \boldsymbol{\Lambda}\mathbf{P} \end{aligned} \tag{25}$$

Note that the leading l_k eigenvectors of Eq. (13) correspond to the columns of the matrix $\mathbf{V} = \mathbf{S}_{w(k)}^{-1} \boldsymbol{\Sigma}$. Therefore, we can prove $\mathbf{F} = \mathbf{V}(\boldsymbol{\Lambda} + \lambda \mathbf{I})^{-1} \boldsymbol{\Lambda}\mathbf{P}$.

According to *Theorem 3*, the eigenvalue problem (13) can be converted to a series of regression problems [33], E and F need to be updated iteratively. If F is fixed, Eq. (22) can be rewritten to

$$\begin{aligned}
&\min_{\mathbf{E}} \sum_{i=1}^{h_k} \left\| \mathbf{S}_{w(k)}^{-T} \mathbf{W}_b^{(k)}(i,:) - \mathbf{E}\mathbf{F}^T \mathbf{W}_b^{(k)}(i,:) \right\|^2 \\
&= \min_{\mathbf{E}} \left\| \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} - \mathbf{W}_b^{(k)} \mathbf{F}\mathbf{E}^T \right\|^2 \\
&= \min_{\mathbf{E}} \mathrm{Tr}\left\{ \left(\mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} - \mathbf{W}_b^{(k)} \mathbf{F}\mathbf{E}^T \right) \left(\mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} - \mathbf{W}_b^{(k)} \mathbf{F}\mathbf{E}^T \right)^T \right\} \\
&= \min_{\mathbf{E}} \mathrm{Tr}\left\{ \left(\mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \mathbf{S}_{w(k)}^{-T} \left(\mathbf{W}_b^{(k)} \right)^T + \mathbf{W}_b^{(k)} \mathbf{F}\mathbf{F}^T \left(\mathbf{W}_b^{(k)} \right)^T \right) \right\} \\
&\quad -2\mathrm{Tr}\left\{ \mathbf{F}^T \left(\mathbf{W}_b^{(k)} \right)^T \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \mathbf{E} \right\}
\end{aligned} \tag{26}$$

If we fix F, the first term in (26) is constant. We only need to maximize $\mathrm{Tr}\{\mathbf{F}^T(\mathbf{W}_b^{(k)})^T \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \mathbf{E}\}$ subject to the constraint $\mathbf{E}^T\mathbf{E} = \mathbf{I}$. The optimal solution can be obtained by resolving the following SVD problem:

$$\mathbf{S}_{w(k)}^{-T} \left(\left(\mathbf{W}_b^{(k)} \right)^T \mathbf{W}_b^{(k)} \right) \mathbf{F} = \mathbf{M}\mathbf{D}\mathbf{N}^T \tag{27}$$

Given a constant F, the optimal solution of (22) is $= \mathbf{M}\mathbf{N}^T$ [41, *Theorem 4*].

Let $\mathbf{E}'$ be an orthogonal matrix, $[\mathbf{E}, \mathbf{E}']$ is also an $h_k \times h_k$ orthogonal matrix that means to concatenate the matrix E and $\mathbf{E}'$ along rows. The first term in (22) is denoted as

$$\begin{aligned}
&\left\| \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} - \mathbf{W}_b^{(k)} \mathbf{F}\mathbf{E}^T \right\|^2 \\
&= \left\| \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} [\mathbf{E}, \mathbf{E}'] - \mathbf{W}_b^{(k)} \mathbf{F}\mathbf{E}^T [\mathbf{E}, \mathbf{E}'] \right\|^2 \\
&= \left\| \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \mathbf{E} - \mathbf{W}_b^{(k)} \mathbf{F} \right\|^2 + \left\| \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \mathbf{E}' \right\|^2 \\
&= \sum_{j=1}^{l_k} \left\| \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \boldsymbol{\mu}_j - \mathbf{W}_b^{(k)} \boldsymbol{\nu}_j \right\|^2 + \left\| \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \mathbf{E}' \right\|^2
\end{aligned} \tag{28}$$

Now, if we fix E, the second term in (28) is constant. The optimal solution F of (28) can be obtained by resolving the following regression problem:

$$\min_{\boldsymbol{\nu}_j} \sum_{j=1}^{l_k} \left\{ \left\| \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \boldsymbol{\mu}_j - \mathbf{W}_b^{(k)} \boldsymbol{\nu}_j \right\|^2 + \lambda \boldsymbol{\nu}_j^T \mathbf{G}_w^{(k)} \boldsymbol{\nu}_j \right\} \tag{29}$$

Here, it is clear that the optimization problem in (29) can be regarded as l_k independent ridge regression problems. Thus, the sparse eigenvectors of (29) can be obtained by resolving these regression problems. However, the solutions of these ridge regression problems are not

always sparse. In order to deal with this problem, we have to impose L1-norm regularization on Eq. (29), and then Eq. (29) can be reformulated to

$$\min_{\boldsymbol{\nu}_j} \sum_{j=1}^{l_k} \left\{ \left\| \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \boldsymbol{\mu}_j - \mathbf{W}_b^{(k)} \boldsymbol{\nu}_j \right\|^2 + \lambda \boldsymbol{\nu}_j^T \mathbf{G}_w^{(k)} \boldsymbol{\nu}_j + \lambda_{1,j} \left\| \boldsymbol{\nu}_j \right\|_1 \right\} \tag{30}$$

Where $\|\boldsymbol{\nu}_j\|_1$ denotes the L1-norm of the vector $\boldsymbol{\nu}_j$, $\lambda_{1,j}$ is the L1-norm tuning coefficient. Therefore, these ridge regression problems have been further converted to a series of LASSO regression problems [3], but they have some drawbacks in practice. For instance, the performance of the extracted features heavily relies on the quantity and quality of training sets. Therefore, we adopt a robust regression algorithm called elastic-net [40] to overcome these drawbacks. The elastic-net combines L1-norm with L2-norm into the objective function. Thus, Eq. (22) can be rewritten as

$$\min_{\mathbf{E},\mathbf{F}} \left(\sum_{i=1}^{h_k} \left\| \mathbf{S}_{w(k)}^{-T} \mathbf{W}_b^{(k)}(i,:) - \mathbf{E}\mathbf{F}^T \mathbf{W}_b^{(k)}(i,:) \right\|^2 + \lambda \sum_{j=1}^{l_k} \boldsymbol{\nu}_j^T \mathbf{G}_w^{(k)} \boldsymbol{\nu}_j + \sum_{j=1}^{l_k} \lambda_{1,j} \left\| \boldsymbol{\nu}_j \right\|_1 + \lambda_2 \left\| \boldsymbol{\nu}_j \right\|^2 \right) \tag{31}$$

The optimization problem in (31) can be resolved by updating E and F iteratively. Further details of the optimization process consist of the following two steps:

(1) F fixed E: For each j, let $f_j^* = \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \boldsymbol{\mu}_j$. Thus, if we fix E, F can be updated by resolving l_k independent elastic-net regression problems, denoted as

$$\min_{\boldsymbol{\nu}_j} \left(\left\| f_j^* - \mathbf{W}_b^{(k)} \boldsymbol{\nu}_j \right\|^2 + \lambda \boldsymbol{\nu}_j^T \mathbf{G}_w^{(k)} \boldsymbol{\nu}_j + \lambda_{1,j} \left\| \boldsymbol{\nu}_j \right\|_1 + \lambda_2 \left\| \boldsymbol{\nu}_j \right\|^2 \right) \tag{32}$$

In order to simplify the objective function of (32), let $f_j = \left(\left(f_j^* \right)^T, \mathbf{0}_{h_k \times h_k} \right)^T$ and $g = \left(\left(\mathbf{W}_b^{(k)} \right)^T, \mathbf{S}_{w(k)}^T \right)^T$. According to the Cholesky decomposition [7], we have $\mathbf{G}_w^{(k)} = \mathbf{S}_{w(k)}^T \mathbf{S}_{w(k)}$. Thus, the elastic-net regression problem (32) are rewritten as

$$\min_{\boldsymbol{\nu}_j} \left(\left\| f_j - g \boldsymbol{\nu}_j \right\|^2 + \lambda_{1,j} \left\| \boldsymbol{\nu}_j \right\|_1 + \lambda_2 \left\| \boldsymbol{\nu}_j \right\|^2 \right) \tag{33}$$

(2) E fixed F: If we fix F and then remove the constant terms from (31), we have

$$\min_{\mathbf{E}} \left(\sum_{i=1}^{h_k} \left\| \mathbf{S}_{w(k)}^{-T} \mathbf{W}_b^{(k)}(i,:) - \mathbf{E}\mathbf{F}^T \mathbf{W}_b^{(k)}(i,:) \right\|^2 \right) = \min_{\mathbf{E}} \left\| \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} - \mathbf{W}_b^{(k)} \mathbf{F}\mathbf{E}^T \right\|^2 \tag{34}$$

The optimal solution is obtained by resolving the following SVD problem:

$$\mathbf{S}_{w(k)}^{-T} \left(\left(\mathbf{W}_b^{(k)} \right)^T \mathbf{W}_b^{(k)} \right) \mathbf{F} = \mathbf{M}\mathbf{D}\mathbf{N}^T \tag{35}$$

According to [41, *Theorem 4*], the optimal solution of (35) is denoted as $\mathbf{E} = \mathbf{M}\mathbf{N}^T$.

Given N–1 sparse projection matrices, i.e. $\mathbf{M}_1, \ldots, \mathbf{M}_{k-1}, \mathbf{M}_{k+1}, \ldots, \mathbf{M}_N$, and then $\mathbf{M}_k$ can be obtained by resolving l_k elastic-net problems. Since $\mathbf{W}_b^{(k)}$ and $\mathbf{S}_{w(k)}^{-1}$ depend on $\{\mathbf{M}_l\}_{l=1,l\neq k}^{N}$, namely, the optimization of the matrix M_k depends on the projections in other tensor modes. The pseudo-code of our RSTSL is summarized in Algorithm 1.

Algorithm 1: Robust Sparse Tensor Subspace Learning

Input: High-order tensor samples $\mathsf{A}_i^p \in \mathbb{R}^{h_1 \times h_2 \times \cdots \times h_N}, i = 1, \ldots, Q_p, p = 1, \ldots, P$, tuning coefficients $\lambda, \lambda_{1,j}, \lambda_2$, the number of dimensions of low-dimensional tensor feature l_k;

Output: Low-dimensional tensor feature $\mathsf{B}_i^p \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_N}, l_i \ll h_i$ and sparse projection matrices $\{\mathbf{M}_k\}_{k=1}^{N}$;

Initialization I: Initialize the projection matrices $\{\mathbf{M}_k\}_{k=1}^{N}$ with a set of identity matrices; Calculate the mean value of images of class p $\bar{\mathsf{A}}_p$ and all classes $\bar{\mathsf{A}}$, respectively; Calculate D_w and D_b through Eq. (15);

For $k = 1$ to N **do**

① Calculate W_w and W_b through Eq. (16);

② Calculate the mode-k unfolded within- and between-class divergence matrix $\mathbf{W}_w^{(k)}$ and $\mathbf{W}_b^{(k)}$, respectively;

③ Calculate the Cholesky decomposition : $\mathbf{G}_w^{(k)} = \mathbf{S}_{w(k)}^T \mathbf{S}_{w(k)}$;

Initialization II: Initialize $\mathbf{E}$ with a set of identity matrices; calculate $\widehat{g} = ((\mathbf{W}_b^{(k)})^T, \mathbf{S}_{w(k)}^T)^T$ and $f_j^* = \mathbf{W}_b^{(k)} \mathbf{S}_{w(k)}^{-1} \mu_j$;

For $j = 1$ to l_k **do**

Calculate $\widehat{f}_j = \left((f_j^*)^T, \mathbf{0}_{h_k \times h_k}\right)^T$;

Optimize $\min_{\mathbf{v}_j} \left(\left\| \widehat{f}_j - \widehat{g}\mathbf{v}_j \right\|^2 + \lambda_{1,j} \left\| \mathbf{v}_j \right\|_1 + \lambda_2 \left\| \mathbf{v}_j \right\|^2 \right)$ through elastic-net regression;

End

Calculate $\mathbf{F}_{h_k \times l_k} = [\mathbf{v}_1, \ldots, \mathbf{v}_{l_k}]$;

Calculate $\mathbf{S}_{w(k)}^{-T} \left((\mathbf{W}_b^{(k)})^T \mathbf{W}_b^{(k)} \right) \mathbf{F} = \mathbf{M}\mathbf{D}\mathbf{N}^T$ by resolving SVD problem;

Calculate $\widehat{\mathbf{E}} = \mathbf{M}\mathbf{N}^T$;

Until $\mathrm{norm}(\mathbf{F}_{m'+1} - \mathbf{F}_{m'}) \leq \Delta'$;

End

Obtain $\mathbf{M}_k = \mathbf{F}$;

Calculate the solution J_{m+1} of (14);

Until $\left| J_{m+1} - J_m \right| < \Delta$;

Calculate low-dimensional tensor feature B_i^p through (5);

Return: Low-dimensional tensor feature $\mathsf{B}_i^p \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_N}, l_i \ll h_i$.

7 3D human pose regression

7.1 Objective function constructing

Until now, we have successfully extracted the LDTF from the high-order tensors. Then, the extracted tensor feature can be used for 3D human pose regression. First, a vector-based linear predictor is introduced, denoted as

$$\mathbf{y} = f(\mathbf{x}; \mathbf{w}, c) = < \mathbf{x}, \mathbf{w} > + c \tag{36}$$

Where x is the input feature vector, y is the regression output vector that is corresponding to x, w is the weight vector and constant c is a bias. In the same way, we introduce a tensor-based predictor

$$\mathbf{y} = f(\mathrm{A}; \mathrm{H}, c) = < \mathrm{A}, \mathrm{H} > + c \tag{37}$$

Where A is the input tensor, H is the weight tensor that is equivalent to the tensor A with mode and dimensionality. Substituting the extracted tensor feature $\mathrm{B} \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_N}, l_i \ll h_i$ into (37), we can have

$$\mathbf{y} = f(\mathrm{B}; \mathrm{H}, c) = < \mathrm{B}, \mathrm{H} > + c \tag{38}$$

To capture the underlying structure information of an image effectively and reduce loss of information, we introduce CP decomposition described in *Definition 4* to represent weight tensor H

$$\mathrm{H} = \sum_{z=1}^{Z} \mathbf{h}_z^{(1)} \circ \mathbf{h}_z^{(2)} \circ \cdots \circ \mathbf{h}_z^{(N)} \triangleq \left(\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, \ldots, \mathbf{H}^{(N)} \right) \tag{39}$$

By concatenating $\mathbf{h}_z^{(j)}$, $j = 1, \ldots, N$ into the matrix $\mathbf{H}^{(j)}$, then we can obtain $\mathbf{H}^{(j)} = [\mathbf{h}_1^{(j)}, \mathbf{h}_2^{(j)}, \ldots, \mathbf{h}_Z^{(j)}]$.

Providing a training sample set $\{(\mathrm{B}_i^p, \mathbf{y}_i^p)\}_{i=1}^n$ of human motion of class p, where $(\mathrm{B}_i^p, \mathbf{y}_i^p)$ is an image-to-pose pair, $\mathrm{B} \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_N}$ is an N - order low-dimensional tensor feature and $\mathbf{y}_i^p$ is the corresponding 3D pose regression vector. The unknown parameter Φ can be estimated only when the empirical risk function below is minimized

$$E(\Phi) = \frac{1}{2} \left(\sum_{i=1}^{n} \ell(\mathbf{y}_i^p, f(\mathrm{B}; \Phi)) + \lambda \Psi(\Phi) \right) \tag{40}$$

Where $\ell(\cdot)$ is a loss function, the over-fitting and complexity are controlled through the regularization term $\Psi(\cdot)$. The unknown parameter Φ is denoted as $\{[\mathbf{H}^{(1)}, \ldots, \mathbf{H}^{(N)}, c]\}$. We introduce a ε-insensitive loss function (ε-ILF) [31], and investigate two types of regularizations (i.e., Frobenius-norm and group-sparsity norm regularization). Thus, the former one leads to HOSTR model while the latter one leads to OOSTR model.

7.2 High-order support tensor regression (HOSTR)

The maximum-margin solution needs to be calculated which is based on the tensor subspace regression. We introduce both the ε-ILF $\ell = \max(0, |\mathbf{y} - f| - \varepsilon)$ and Frobenius-norm

regularization $\Psi_1(\Phi) = ||\mathrm{H}||_{\mathrm{F}}^2$ into our proposed objective function (see Eq. (40)). Thus, we can obtain the HOSTR model [10], denoted as

$$\begin{cases} \min\limits_{\mathbf{H},c,\phi,\hat{\phi}} \quad \lambda\sum\limits_{i=1}^{n}\left(\phi_i+\hat{\phi}_i\right)+\frac{1}{2}\|\mathrm{H}\|_{\mathrm{F}}^2 \\ s.t. \quad -\mathbf{y}_i^p+<\mathrm{B}_i^p,\mathrm{H}>+c\geq\varepsilon+\hat{\phi}_i, \\ \quad\ \mathbf{y}_i^p-<\mathrm{B}_i^p,\mathrm{H}>-c\leq\varepsilon+\phi_i, \\ \quad\ \varepsilon\geq 0,\phi_i\geq 0,\hat{\phi}_i\geq 0, i=1,2,\ldots,n \end{cases} \tag{41}$$

To optimize (41) with respect to $\mathbf{H}^{(j)}$, we fix all $\mathbf{H}^{(z)}|_{z=1}^{N}$ except $z=j$, then, we can have

$$\begin{cases} \min\limits_{\mathbf{H}^{(j)},c,\phi,\hat{\phi}} \quad \lambda\sum\limits_{i=1}^{n}\left(\phi_i+\hat{\phi}_i\right)+\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{(j)}\left(\mathbf{H}^{(-j)}\right)^T\mathbf{H}^{(-j)}\left(\mathbf{H}^{(j)}\right)^T\right) \\ s.t. \quad -\mathbf{y}_i^p+\mathrm{Tr}\left(\mathbf{H}^{(j)}\left(\mathbf{H}^{(-j)}\right)^T\mathbf{B}_{(j)i}^T\right)+c\geq\varepsilon+\hat{\phi}_i, \\ \quad\ \mathbf{y}_i^p-\mathrm{Tr}\left(\mathbf{H}^{(j)}\left(\mathbf{H}^{(-j)}\right)^T\mathbf{B}_{(j)i}^T\right)-c\leq\varepsilon+\phi_i, \\ \quad\ \varepsilon\geq 0,\phi_i\geq 0,\hat{\phi}_i\geq 0, i=1,2,\ldots,n \end{cases} \tag{42}$$

We denote $\Theta=\left(\mathbf{H}^{(-j)}\right)^T\mathbf{H}^{(-j)}$, $\mathbf{H}^{(j)}=\mathbf{H}^{(j)}\sqrt{\Theta}$ and $\mathbf{B}_{(j)i}=\mathbf{B}_{(j)i}\mathbf{H}^{(-j)}\sqrt{\Theta}$, then, we have

$$\begin{cases} \min\limits_{\mathbf{H}^{(j)},c,\phi,\hat{\phi}} \quad \lambda\sum\limits_{i=1}^{n}\left(\phi_i+\hat{\phi}_i\right)+\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{(j)}\left(\mathbf{H}^{(j)}\right)^T\right) \\ s.t. \quad -\mathbf{y}_i^p+\mathrm{Tr}\left(\mathbf{H}^{(j)}\mathbf{B}_{(j)i}^T\right)+c\geq\varepsilon+\hat{\phi}_i, \\ \quad\ \mathbf{y}_i^p-\mathrm{Tr}\left(\mathbf{H}^{(j)}\mathbf{B}_{(j)i}^T\right)-c\leq\varepsilon+\phi_i \\ \quad\ \varepsilon\geq 0,\phi_i\geq 0,\hat{\phi}_i\geq 0, i=1,2,\ldots,n \end{cases} \tag{43}$$

We convert the matrix $\mathbf{H}^{(j)}$ and $\mathbf{B}_{i(j)}^T$ into their corresponding vectors, denoted as

$$\begin{cases} \mathrm{Tr}\left(\mathbf{H}^{(j)}\left(\mathbf{H}^{(j)}\right)^T\right)=\left\|\mathrm{vec}\left(\mathbf{H}^{(j)}\right)\right\|^2 \\ \mathrm{Tr}\left(\mathbf{H}^{(j)}\mathbf{B}_{(j)i}^T\right)=\left[\mathrm{vec}\left(\mathbf{H}^{(j)}\right)\right]^T\cdot\left[\mathrm{vec}\left(\mathbf{B}_{(j)i}^T\right)\right] \end{cases} \tag{44}$$

The optimization problem in (43) can be solved by using a classical SVM optimizer [28, 31]. Once $\mathbf{H}^{(j)}$ is calculated, we can also calculate $\mathbf{H}^{(j)}$ according to

$$\mathbf{H}^{(j)} = \mathbf{H}^{(j)} / \sqrt{\Theta} \tag{45}$$

7.3 Optimal-order support tensor regression (OOSTR)

Although a low complexity model can be obtained by realizing a lower error bound. Nevertheless, a problem that aims to optimize tensor decomposition order is NP-hard. In order to bypass this problem and automatically learn the optimal tensor order in the regression process, we use a group-sparsity norm regularization to achieve it instead of conventional Frobenius-norm regularization

$$\Psi(\mathrm{H}) = \sum_{z=1}^{Z} \sqrt{\left(\sum_{k=1}^{N} \left\| \mathbf{H}_{:,z}^{(k)} \right\|_2^2 \right)} \tag{46}$$

Where $\mathbf{H}_{:,z}^{(k)}$ denotes the zth column of the matrix $\mathbf{H}^{(k)}$, therefore, we can force the zth column of all $\{\mathbf{H}^{(j)}\}_{j=1}^{N}$ to be zero simultaneously using this regularization.

Lemma 2 The group-sparsity norm regularization $\Psi(\mathrm{H})$ in Eq. (46) can be converted to a minimization problem:

$$\Psi(\mathrm{H}) = \sum_{z=1}^{Z} \sqrt{\left(\sum_{k=1}^{N} \left\| \mathbf{H}_{:,z}^{(k)} \right\|_2^2 \right)} = \min_{\boldsymbol{\varphi} \in \mathbb{R}^Z} \frac{1}{2} \sum_{z=1}^{Z} \frac{\sum_{k=1}^{N} \left\| \mathbf{H}_{:,z}^{(k)} \right\|_2^2}{\varphi_z} + \frac{1}{2} \|\boldsymbol{\varphi}\|_1 \tag{47}$$

Where $\varphi_z = \sqrt{\left(\sum_{k=1}^{N} \left\| \mathbf{H}_{:,z}^{(k)} \right\|_2^2 \right)}$, z = 1,…,Z is the closed-form solution of (47)

Proof Let $u, v \geq 0$, then we can have $(u-v)^2 \geq 0$ i.e. $u \leq u^2/2v + v/2$. Let $u^2/v = 0$ when $u = 0, v = 0$ and otherwise $u^2/v = +\infty$ when $v = 0, u \neq 0$. Thus the equality holds for $u = v$, then

$$\|\boldsymbol{q}\|_1 = \sum\nolimits_z |q_z| \leq \sum\nolimits_z \frac{1}{2} \left(\frac{q_z^2}{v_z} + v_z \right) = \sum\nolimits_z \frac{q_z^2}{2v_z} + \frac{1}{2} \|v\|_1 \tag{48}$$

Let $q_z = \sqrt{\left(\sum_{k=1}^{N} \left\| \mathbf{H}_{:,z}^{(k)} \right\|_2^2 \right)}$ and $\eta_z = v_z$, substituting them into (48), we can prove (47).

According to *Lemma 2*, we present an OOSTR model, and further details of our model are described in Algorithm 2. Substituting (46) into (40), we can obtain the optimal-order tensor learning model:

$$\begin{aligned}\min_{\mathbf{H}^{(k)}\big|_{k=1}^{N},c,\boldsymbol{\eta}} \quad & E\left(\mathbf{H}^{(k)}\big|_{k=1}^{N},c,\boldsymbol{\varphi}\right)\\ &=\frac{1}{2}\sum_{i=1}^{N}\ell\left(\mathbf{y}_i^p,\left\langle \mathrm{B}_i^p,\left(\mathbf{H}^{(1)},\mathbf{H}^{(2)},\ldots,\mathbf{H}^{(N)}\right)\right\rangle+c\right)\\ &+\frac{\lambda}{4}\left(\sum_{z=1}^{Z}\frac{\sum_{k=1}^{N}\left\|\mathbf{H}_{:,z}^{(k)}\right\|_2^2}{\varphi_z}+\|\boldsymbol{\varphi}\|_1\right)\end{aligned} \tag{49}$$

Now, we adopt a block-based coordinate-descent method [23], and its analytical form is following:

$$\begin{cases}\left(\mathbf{H}^{(j)(z+1)},c^{z+1}\right)\leftarrow\arg\min_{\mathbf{H}^{(j)},c}E\left(\mathbf{H}^{(j)},\mathbf{H}^{(k)(z)}\big|_{k=1,k\neq j}^{N},\boldsymbol{\varphi}^{(z)}\right)\\ \boldsymbol{\varphi}^{(z+1)}\leftarrow\arg\min_{\boldsymbol{\varphi}}E\left(\boldsymbol{\varphi},\mathbf{H}^{(k)(z)}\big|_{k=1}^{N},c^{(z)}\right)\end{cases} \tag{50}$$

Given the rest of N–1 matrices $\{\mathbf{H}^{(k)}|_{k=1,k\neq j}^{N},c\}$, φ is updated by the closed-form solutions provided by *Lemma 2*

$$\varphi_z=\sqrt{\left(\sum_{k=1}^{N}\left\|\mathbf{H}_{:,z}^{(k)}\right\|_2^2\right)}+\varepsilon,\quad z=1,\ldots,Z \tag{51}$$

To avoid the problem of numeric ambiguity, we fix $(\mathbf{H}^{(k)})|_{k=1,k\neq j}^{N}$ subject to the constraint $0<\varepsilon\ll 1$. Thus, we can obtain the following sub-problems with respect to $\mathbf{H}^{(j)}$, denoted as

$$E_j\left(\mathbf{H}^{(j)},c\right)=\frac{1}{2}\sum_{i=1}^{n}\ell\left(\mathbf{y}_i^p,\mathrm{Tr}\left(\mathbf{H}^{(j)}\mathbf{H}^{(-j)^T}\mathbf{B}_{(j)i}^T\right)+c\right)+\frac{\lambda}{4}\mathrm{Tr}\left(\mathbf{H}^{(j)}\boldsymbol{\Lambda}\left(\mathbf{H}^{(j)}\right)^T\right) \tag{52}$$

Where $H^{(-j)}=H^{(N)}\odot\cdots\odot H^{(j+1)}\odot H^{(j-1)}\odot\cdots\odot H^{(1)}$ *and* $\Lambda=diag(1/\varphi_1,1/\varphi_2,\ldots,1/\varphi_z)$. The operator "⊙" denotes Khatri-Rao product. We use both the ε-ILF $\ell(\cdot)$ and the

group-spasity norm regularization $\Psi(\cdot)$ in the optimization problem. The sub-problem in (52) can be then reformulated to

$$\begin{cases} \min\limits_{\mathbf{M}^{(j)},c,\phi} & \lambda\sum\limits_{i=1}^{n}\left(\phi_i+\hat{\phi}_i\right)+\frac{1}{2}Tr\left(\mathbf{H}^{(j)}\mathbf{\Lambda}\left(\mathbf{H}^{(j)}\right)^T\right) \\ s.t. & -\mathbf{y}_i^p+\mathrm{Tr}\left(\mathbf{H}^{(j)}\left(\mathbf{H}^{(-j)}\right)^T\mathbf{B}_{(j)i}^T\right)+c\geq\varepsilon+\hat{\phi}_i, \\ & \mathbf{y}_i^p-\mathrm{Tr}\left(\mathbf{H}^{(j)}\mathbf{H}^{(-j)}\mathbf{B}_{(j)i}^T\right)-c\leq\varepsilon+\phi_i, \\ & \varepsilon\geq 0,\ \phi_i\geq 0,\ \hat{\phi}_i\geq 0,\ i=1,\ldots,n \end{cases} \tag{53}$$

The Eq. (53) is equivalent to Eq. (43) if we replace Θ with $\mathbf{\Lambda}$, therefore, the optimization problem (53) can also be resolved using the same strategy as section 7.2.

Algorithm 2: 3D human pose regression

Input: A training sample set $\{(\mathsf{B}_i^p,\mathbf{y}_i^p)\}_{i=1}^n$, where $(\mathsf{B}_i^p,\mathbf{y}_i^p)$ denotes an image-to-pose pair, and n denotes the total number of training samples, the largest number of iterations T_{max};

Output: the weight tensor H and the bias c;

Start: To estimate 3D human pose from an unknown input tensor, undoubtedly, we need to calculate the weight tensor H and the bias c. Thus, three steps are needed for the 3D human pose regression:

① Initialize $\{\mathbf{H}^{(1)},\mathbf{H}^{(2)},...,\mathbf{H}^{(N)}\}^{(0)}$;

② $\tau\leftarrow\tau+1$;

③ **For** $k=1$ to N **do**

Optimize with respect to $\mathbf{H}^{(k)}|(\tau)$;

For OOSTR model solve (53);

End

Update parameter φ using the closed-form solutions provided by ***Lemma2***;

Crop the columns of $\mathbf{H}_{:z}^{(k)}$ subject to the constraint $k\in\{1,\ldots,N\},\ z\in\{j\mid\varphi_j\leq\varepsilon,j=1,\ldots,Z\}$;

Until $\left\|\mathsf{H}^{(\tau)}-\mathsf{H}^{(\tau-1)}\right\|/\left\|\mathsf{H}^{(\tau-1)}\right\|\leq\varepsilon$ or $\tau\geq T_{max}$;

Obtain $\mathsf{H}^*=\{\mathbf{H}^{(1)},\mathbf{H}^{(2)},...,\mathbf{H}^{(N)}\}$ and c^*;

Return: 3D human pose configuration $\mathbf{y}_i^p=f(\mathsf{B}_i^p;\mathsf{H}^*,c^*)=<\mathsf{B}_i^p,\mathsf{H}^*>+c^*$.

7.4 Convergence analysis

From [22, 30], the value of function $E(\{\mathbf{H}^{(1)},\ldots,\mathbf{H}^{(N)}\},c,\boldsymbol{\varphi})$ is invariant to $\mathrm{H}^{(l)}$ between two continuous iterations. Now, we constrain the function E to be a continuous form, denoted as $E:\mathbf{H}_1\times\cdots\times\mathbf{H}_N\times\mathbb{R}^n\times\mathbb{R}=\prod_{l=1}^N\mathbf{H}^{(l)}\times\mathbb{R}^n\times\mathbb{R}\rightarrow\mathbb{R}^+$, where $\mathbf{H}^{(j)}\in\mathbb{N}_j\in\mathbb{R}^{I_j\times V}$, $c\in\mathbb{R}$. Thus, the form of function E can be reduced to $E:\mathbb{N}_j\times\mathbb{R}^n\times\mathbb{R}\rightarrow\mathbb{R}$. Then, we define a continuous function $E(\mathbf{H}^{(j)},c,\boldsymbol{\varphi})=E(\mathbf{H}^{(j)},c,\boldsymbol{\varphi};\mathbf{H}^{(k)}(\tau)|_{k=1}^{j-1},\mathbf{H}^{(k)}|_{k=j+1}^N)$. If we fix $H^{(i)}|_{i\neq j}^N$, the function E is optimized with respect to $\mathbf{H}^{(j)}$. The symbol "*" denotes the optimal solutions. Then the N mappings of function E are following

$$\psi\left(\mathbf{H}_*^{(j)},c_*^{(j)},\phi\right)\triangleq\arg\min_{\mathbf{H}^{(j)},c}E\left(\mathbf{H}^{(k)}\Big|_{k=1}^N,c,\phi\right)=\arg\min_{\mathbf{H}^{(j)},c}E\left(\mathbf{H}^{(j)},c\right) \tag{54}$$

Where ϕ is an additional set of parameters, and according to our OOSTR, we have $\psi\left(\mathbf{H}_*^{(j)}, c_*^{(j)}, \phi\right) = \mathbf{H}^{(j)}/\sqrt{\mathbf{\Lambda}}$, then

$$E\left(\mathbf{H}_*^{(j)}, c_*^{(j)}, \phi\right) \geq E\left(\mathbf{H}^{(1)}, c^{(j)}, \phi\right) \tag{55}$$

Given an initialization $\mathbf{H}^{(k)}(0)|_{k=1}^{N}$, Algorithm 2 can produce the optimal solutions $\{\mathbf{H}_*^{(k)}(\tau)|_{k=1}^{N}, c_*^{(j)}(\tau), \phi\}$ via

$$\psi\left(\mathbf{H}_*^{(j)}(\tau), c_*^{(j)}(\tau), \phi\right) \triangleq \arg\min_{\mathbf{H}^{(j)}, c} E\left(\mathbf{H}^{(j)}, c\right), j = \{1, \ldots, N\} \tag{56}$$

The optimal solutions produced by Algorithm 2 can be characterized by the following inequalities:

$$\begin{aligned}
\boldsymbol{\rho}_1 &= E\left(\mathbf{H}_*^{(1)}(1), c_*^{(1)}(1), \phi\right) \\
&\geq E\left(\mathbf{H}_*^{(2)}(1), c_*^{(2)}(1), \phi\right) \geq \cdots \geq E\left(\mathbf{H}_*^{(N)}(1), c_*^{(N)}(1), \phi\right) \\
&\geq E\left(\mathbf{H}_*^{(1)}(2), c_*^{(1)}(2), \phi\right) \geq \cdots \geq E\left(\mathbf{H}_*^{(1)}(\tau), c_*^{(1)}(\tau), \phi\right) \\
&\geq E\left(\mathbf{H}_*^{(2)}(\tau), c_*^{(2)}(\tau), \phi\right) \\
&\geq E\left(\mathbf{H}_*^{(1)}(T), c_*^{(1)}(T), \phi\right) \geq \cdots \geq E\left(\mathbf{H}_*^{(N)}(T), c_*^{(N)}(T), \phi\right) \geq \boldsymbol{\rho}_2
\end{aligned} \tag{57}$$

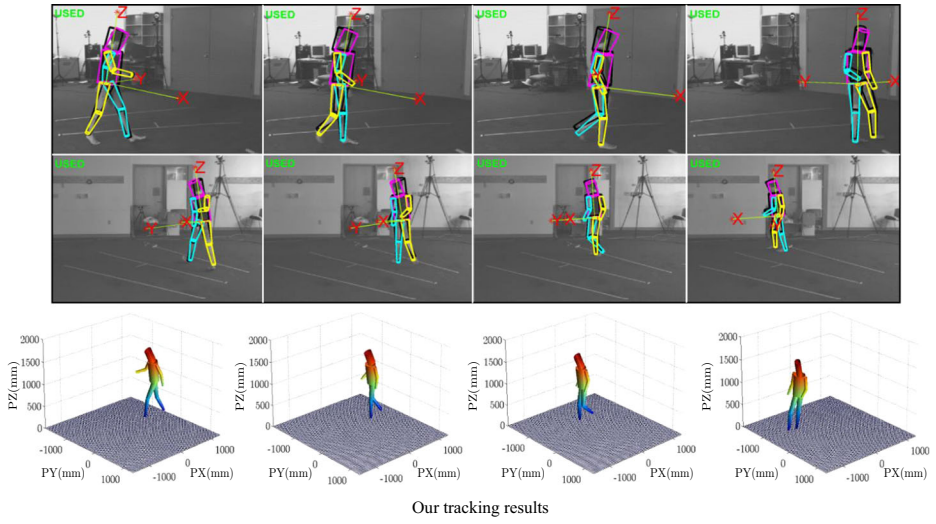

Fig. 6 Sample tracking results by all algorithms at frame #153, #169, #205 and #251 under the condition of sudden change in human velocity. First row: camera one. Second row: camera three. Third row: the corresponding 3D human model (**a**) Our tracking results (**b**) TGP tracking results (**c**) APF tracking results

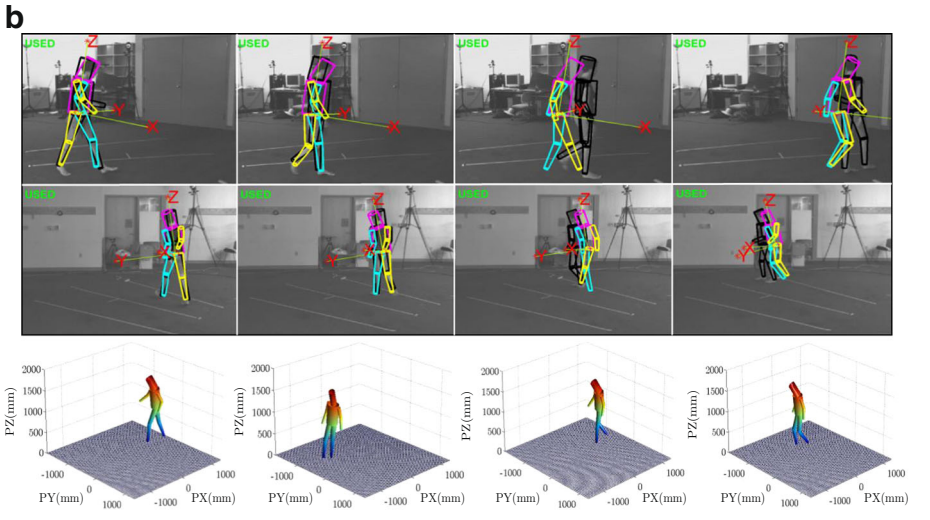

Fig. 6 continued.

Where $T \to \infty, \rho_1$ and $\boldsymbol{\rho}_2$ are limited values in $\mathbb{R}$, thus, the iterative optimization can be regarded as a combinational optimization problem formed by a set of sub-algorithms as described in Eq. (58)

$$\omega^{(j)} : \left(\mathbf{H}^{(k)} \Big|_{k=1}^{N}, \phi, c\right) \to \mathbb{R}^{l_1 \times V} \times \cdots \times \mathbb{R}^{l_j \times V} \times \mathbb{R} \tag{58}$$

The $\mathbf{H}^{(j)}$ and c can be obtained via Eq. (58), and $\omega = \omega_1 \circ \omega_2 \circ \cdots \circ \omega_N = \circ_{d=1}^{N} \omega_d$ is closed when all $\mathbb{N}$ are compact. Since all sub-algorithms decrease the value of function g, so it is clear that ω is monotonic with respect to function g. Therefore, we can conclude that the alternating projection algorithm converges.

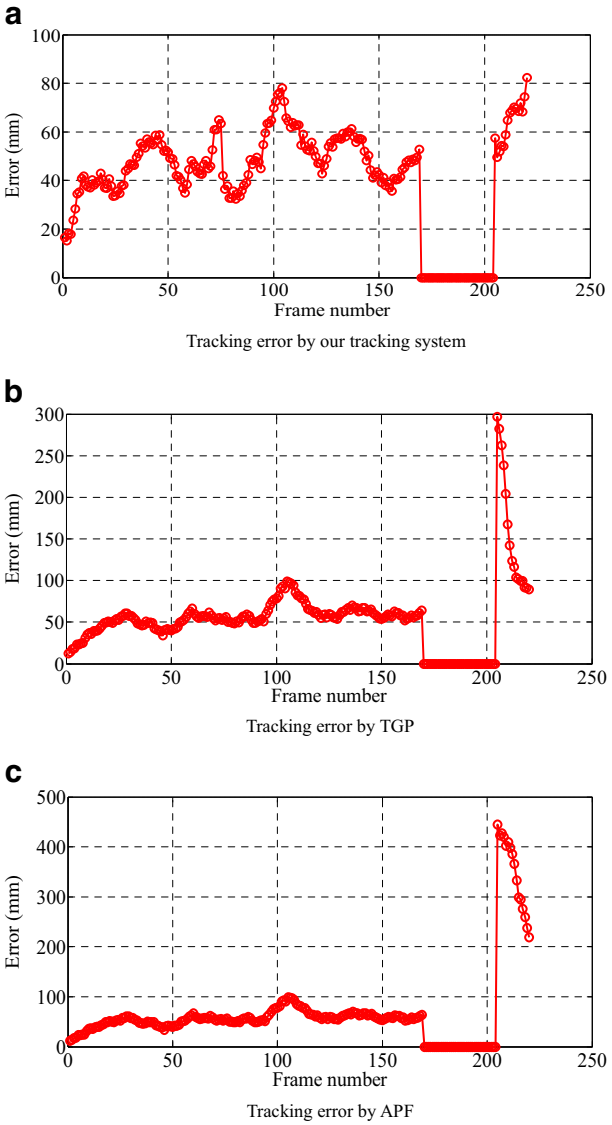

Fig. 7 Tracking errors by all algorithms under the condition of sudden change in human velocity (**a**) Tracking error by our tracking system (**b**) Tracking error by TGP (**c**) Tracking error by APF

8 Experimental results analysis

In this section, we introduce two publicly available human motion databases (i.e. Brown [26] database and HumanEva database [27]) that are conducted in our experiments: (1) The circular *Walking* sequences from Brown database (4-gray views, i.e. G1 ~ G4); (2) 4 subjects and 5 activities from HumanEva-I database (3-color views, i.e. C1 ~ C3). Additionally, the combo motion sequences from HumanEva-II database (4-color views, i.e. C1 ~ C4).

8.1 Brown database

The Brown database involves 4 types of files: (1) 4-view gray images; (2) human motion capture data; (3) binary maps for foreground silhouettes; (4) camera calibration files. The gray image sequences from Brown database are taken by 4 synchronized cameras at 60 Hz. The researchers utilize a Vicon system to collect human motion capture data which is based on 6 cameras at 120 Hz. The collected capture data is processed to get human body model

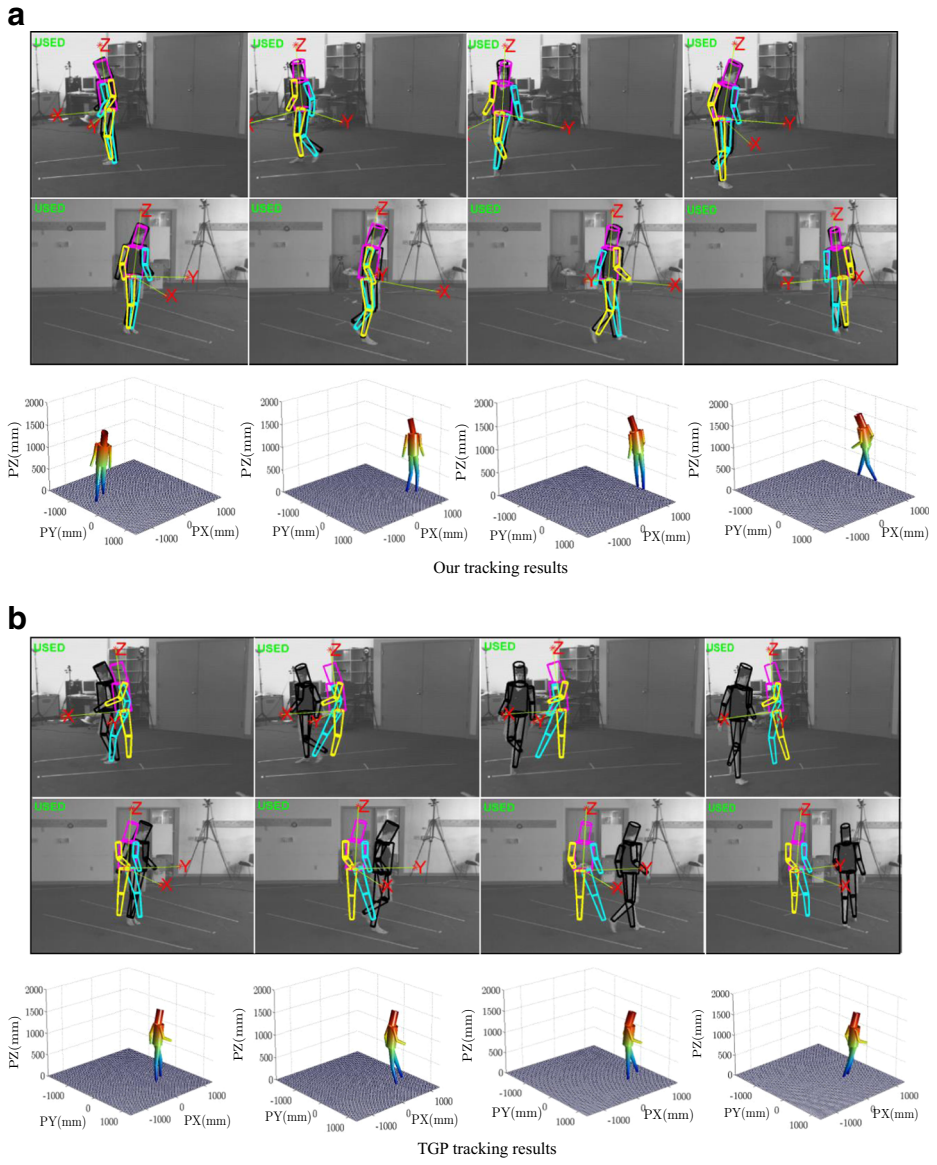

Fig. 8 Sample tracking results by all algorithms at frame #30, #60, #90 and #120 under the condition of low-frame rate. First row: camera one. Second row: camera three. Third row: the corresponding 3D human model (**a**) Our tracking results (**b**) TGP tracking results (**c**) APF tracking results

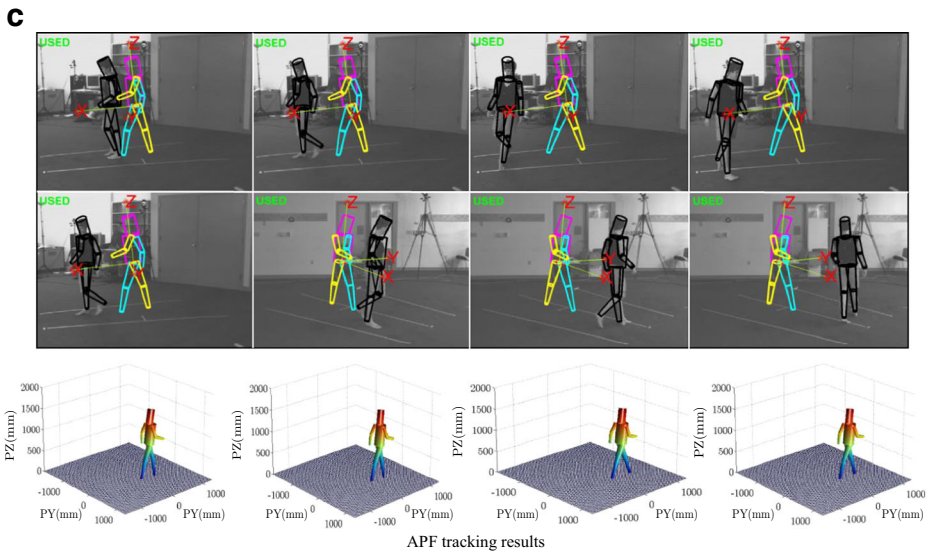

Fig. 8 continued.

parameters containing angles, locations and orientations of joint centers. The 3D human model from this database has 10 parts containing head, torso, upper and lower arms (right and left), thighs and calves (right and left) [26]. The 4 gray cameras are employed in our experiment, and we evaluate all algorithms from two aspects, i.e., sudden change in human velocity and low-frame rate. We draw the ground-truth pose with black lines and the estimated pose with magenta, cyan and yellow lines.

8.1.1 Sudden change in human velocity

We test the performance of our tracking system, Twin Gaussian Process (TGP) [2] and APF [4] in the case of sudden change in human velocity. The results are illustrated in Fig. 6. Both TGP and APF are very representative and effective methods in the past few years, and the two are all based on the vector-based space. Therefore, they are in sharp with our proposed pipeline that is mainly based on the tensor-based space, which is beneficial to confirm both the efficiency and superiority of our tensor-based pipeline when compared with those vector-based ones. Specifically, APF is applied to a large amount of pose tracking applications. Finally, TGP and OOSTR all belong to the regression class. Their tracking accuracy highly relies on the feature representation, which is also convenient to verify the robustness of our RSTSL.

In our experiments, we skip frame Nos.170 ~ 204 and then sudden change in human velocity happens between frame No. 169 and frame No. 205. Figure 6a shows part of tracking results by our tracking system. It can be shown that our proposed algorithm can reconstruct the 3D human pose from the 2D visual observation accurately and effectively in such condition. By contrast, TGP and APF fail to complete this task (see Fig. 6b, c), and there exists an increasing mismatch between the limbs of human model and those of foreground.

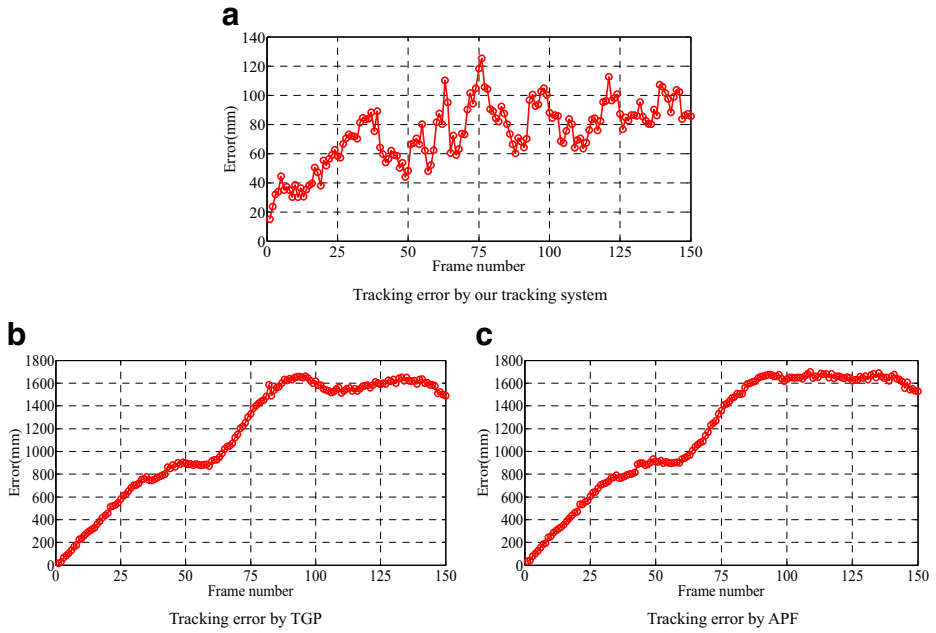

Fig. 9 Tracking errors by all algorithms under the condition of low-frame rate (**a**) Tracking error by our tracking system (**b**) Tracking error by TGP (**c**) Tracking error by APF

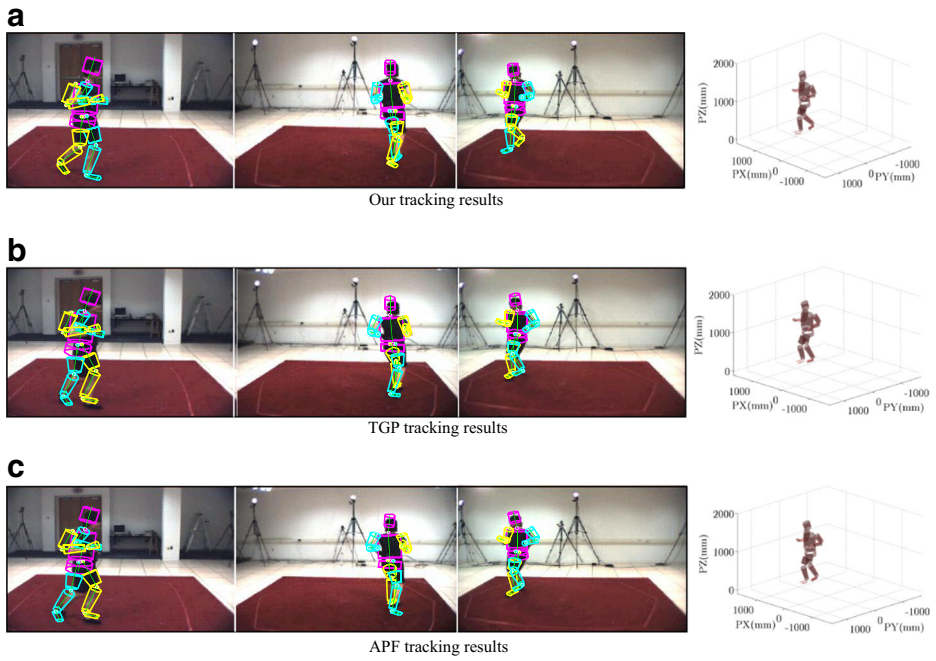

Fig. 10 The 3-view tracking results of frame #354 of S1 performing *Jogging* are used to test the performance of disambiguating silhouette

In this paper, we calculate the tracking errors using the similar evaluation strategy presented in [27]. The mean error (in millimeter) over all joint centers is defined as

$$Err(\mathbf{y}', \mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N} \|f_i(\mathbf{y}') - f_i(\mathbf{y})\|,\ f_i(\mathbf{y}'), f_i(\mathbf{y}) \in \mathbb{R}^3 \tag{59}$$

Where $f_i(\cdot)$ denotes a function that is used to extract the 3D locations of body joint centers, and N denotes the total number of body joint centers for each human pose. Operator "$\|\cdot\|$" is Euclidean distance between the 3D locations of corresponding joins. In addition, $\mathbf{y}' = (\mathbf{y}'_1, \ldots, \mathbf{y}'_N)$ and $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$ are the estimated pose and the ground-truth. Figure 7 plots the errors by our tracking system, TGP and APF under the condition of sudden change in human velocity.

It can be shown that the tracking errors by all algorithms are less than 100 mm before frame No. 170. However, after frame No. 204 when the sudden change in human velocity happens, the error by TGP increases to around 300 mm and the error by APF is even more than 400 mm while the error by our tracking system is still less than 100 mm.

8.1.2 Low-frame rate

In this section, we test all algorithms when the frame rate is down to 15 fps. To get the low-frame rate image sequences, the original image sequences from Brown database need to be re-sampled every 4 frames. The tracking results by our tracking system are illustrated in Fig. 8a, it can be shown that our tracking system can still track the 3D human poses from the low frame

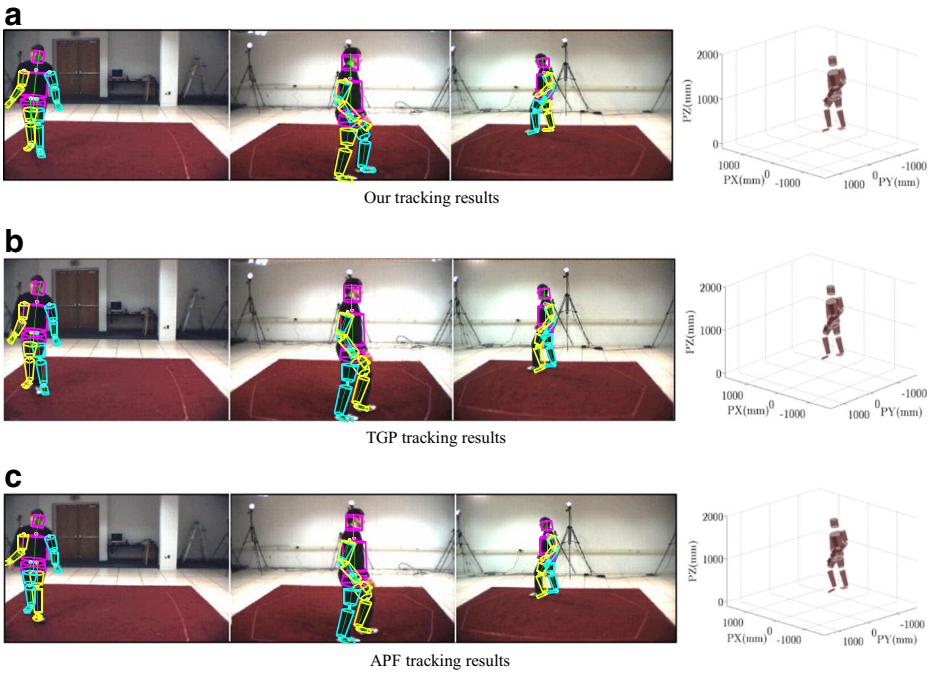

Fig. 11 The 3-view tracking results of frame #296 of S3 performing *Walking* are used to test the performance of disambiguating silhouette

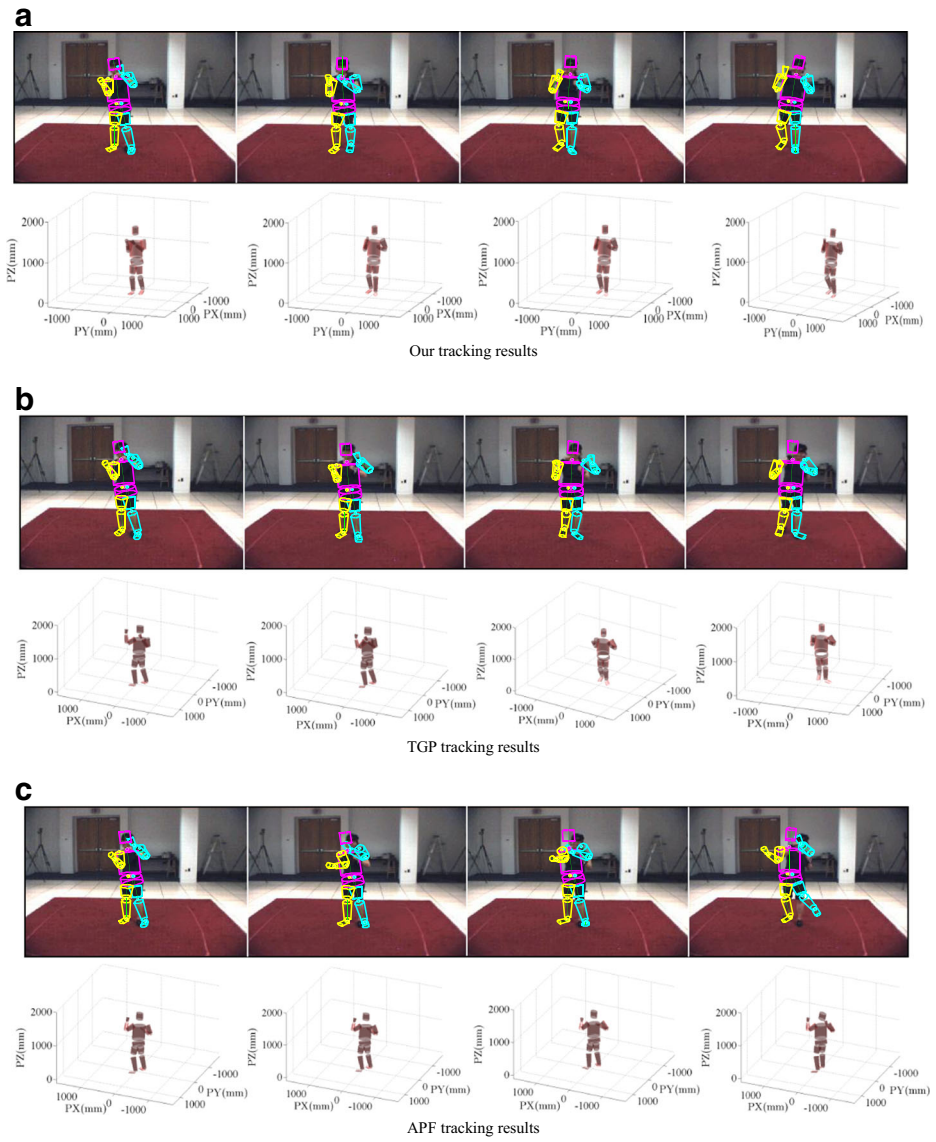

Fig. 12 Tracking results of frame #112, #214, #282 and #507 of S1 performing *Boxing* are used to test the performance of overcoming transient occlusion. First row: the estimated pose. Second row: the corresponding 3D human model

rate image sequences satisfactorily. However, TGP and APF fail to complete the tracking tasks and even thoroughly lose the human body target (see Fig. 8b and c).

Figure 9 reports the tracking errors over 150 frames by our tracking system, TGP and APF (see Fig. 9a, b, and c). It can be shown that the tracking error by our tracking system is always less than about 130 mm while the tracking error by TGP increases to around 1600 mm and tracking error by APF even increases to around 1700 mm.

8.2 HumanEva database

The HumanEva database is taken by 7 cameras (i.e. 3-color and 4-gray cameras) or 4-color cameras and a Vicon motion capture system. All videos and image sequences from this database are synchronized. A group of different subjects are required to perform some given

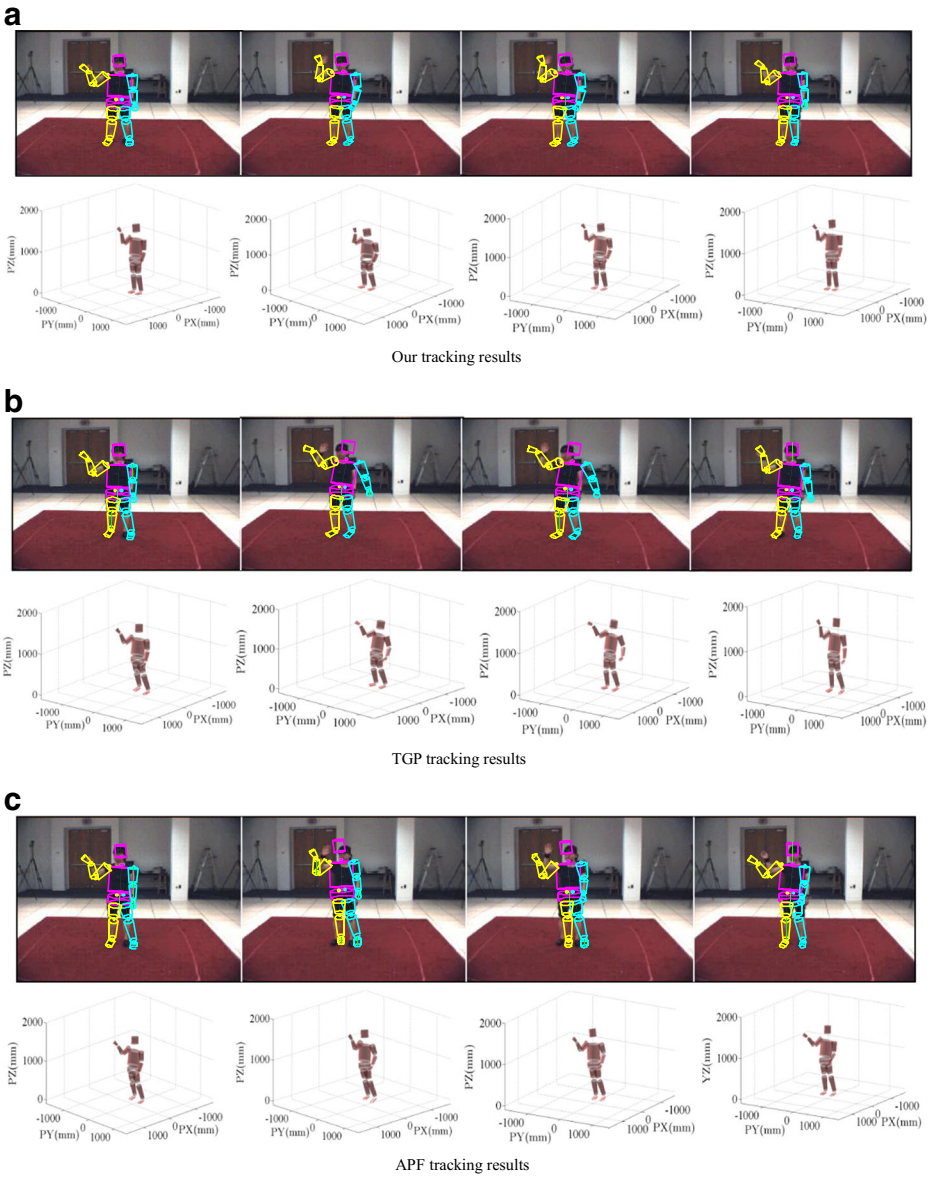

Fig. 13 Tracking results of frame #184, #312, #391 and #520 of S1 performing *Gestures* are used to test the performance of overcoming transient occlusion. First row: the estimated pose. Second row: the corresponding 3D human model

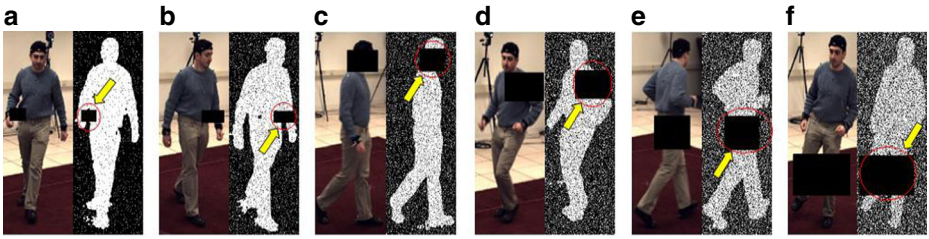

Fig. 14 Partial test images from HumanEva-II with 6 different corruption levels (**a**) CL1 (5 %) (**b**) CL2 (15 %) (**c**) CL3 (25 %) (**d**) CL4 (35 %) (**e**) CL5 (45 %) (**f**) CL6 (55 %)

motions periodically, including *Walking*, *Jogging*, *Boxing*, *Gestures*, and *Throw/Catch*. The ground-truth of this database provides the 3D locations of body joints. In total, there are 15 limbs containing head, torso, pelvis, upper and lower arms (right and left), thighs and calves (right and left), feet and hands (right and left) [27]. Moreover, the HumanEva database is portioned into three subsets, i.e. training, testing and validation. Nevertheless, the ground-truth of testing subset is not given. In order to make up for this shortcoming, we consider the original validation subset as testing data and the original training subset as training data. Finally, a total number of 2950 frames (i.e. trial 1 of S1, S2 and S3) for *Walking*, 2345 frames for *Jogging*, 2850 frames for *Gestures*, 2486 frames for *Boxing* and 2367 frames for *Throw/Catch* are utilized. Moreover, the 3-view visual information we utilize is based on camera C1 ~ C3. We evaluate all algorithms on HumanEva-I database from two points, i.e. eliminating silhouette ambiguity and overcoming transient occlusion of cameras. Finally, we also test the capability of our tracking system to deal with obstacle occlusion and random noise disturbance on HumanEva-II database.

8.2.1 Eliminating the silhouette ambiguity

We first test the performance of our tracking system to eliminate silhouette ambiguity on HumanEva-I database. A 2D human silhouette is inherently ambiguous. For instance, we cannot determine which leg of a 2D human silhouette is in front or in back. Additionally, we also cannot accurately determine 2D pixel coordinates of arms or other body limbs in the case of self-occlusion. Therefore, the ambiguity often occurs since a 2D human silhouette is always associated with more than one 3D human pose. In our experiments, 5 types of activities are chosen as our test cases, including *Walking*, *Jogging*, *Boxing*, *Gestures* and *Throw/Catch*. The 3-view results of S1 performing *Jogging* by our tracking system, TGP and APF are shown in Fig. 10, and those of S3 performing *Walking* by all algorithms are shown in Fig. 11.

Figures 10a and 11a show the 3-view tracking results estimated by our tracking system. We choose the frame No. 354 from test video of S1 performing *Jogging* and the frame No. 296 from test video of S3 performing *Walking* as our test cases. Furthermore, 3-color cameras are utilized in our experiments.

The experimental results show that our tracking system can not only track the 3D human poses from the multi-view images accurately but also eliminate the silhouette ambiguity caused by self-occlusion effectively. However, TGP and APF fail to deal with this silhouette ambiguity problem, and there exists a mismatching between the limbs of 3D human model and those of foreground (see Figs. 10b and c, 11b and c). Additionally, the correct colors of left and

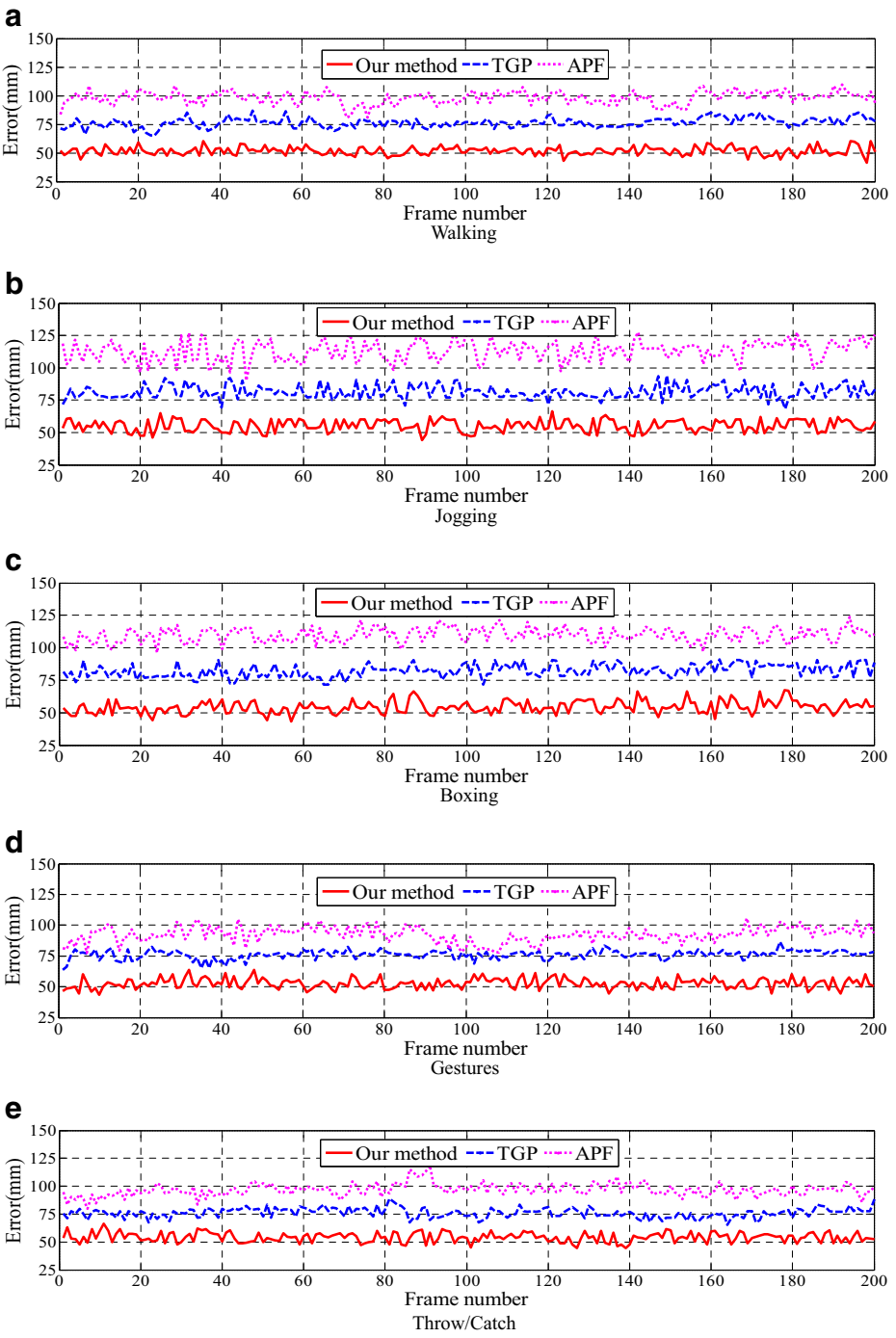

Fig. 15 Tracking errors by our method, TGP and APF of S1 performing 5 kinds of activities (**a**) Walking (**b**) Jogging (**c**) Boxing (**d**) Gestures (**e**) Throw/Catch

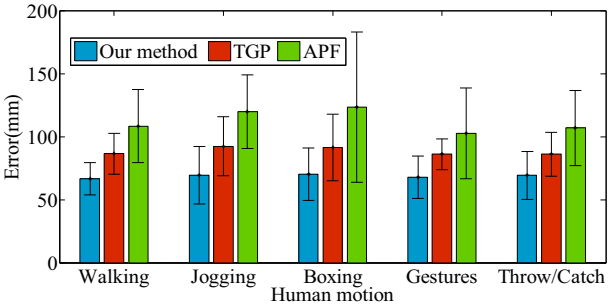

Fig. 16 The mean and standard deviation of tracking errors

right leg should be cyan and yellow, but those of the tracked poses by TGP and APF are opposite.

8.2.2 Overcoming transient occlusion of cameras

The problem of transient occlusion of cameras, in general how to recover the incoherent human motions, usually results in bad silhouettes or silhouette missing from all views. In the following experiments, we choose 4 discontinuous frames from test video of S1 performing *Boxing* and another 4 discontinuous frames from test video of S1 performing *Gestures* to evaluate all algorithms. The tracking results by our tracker, TGP and APF are illustrated in Figs. 12 and 13. It can be shown that our tracker achieves the most significant performance among all algorithms, and finally all the estimated poses are projected to the camera C1 view.

The experimental results show that our tracking system can not only effectively overcome the transient occlusion but also accurately estimate the 3D human poses. By contrast, TGP and APF fail to overcome this problem and estimate the human poses. In Fig. 12b, c, there exists a larger mismatching between the calves and left arm of the human model and those of foreground from camera C1. In Fig. 13b, c, there exists a larger mismatching between the right arm of the human model and that of foreground from camera C1. Figure 15 shows the tracking errors by all algorithms, which are corresponding to the tracking results of S1 performing 5 kinds of activities.

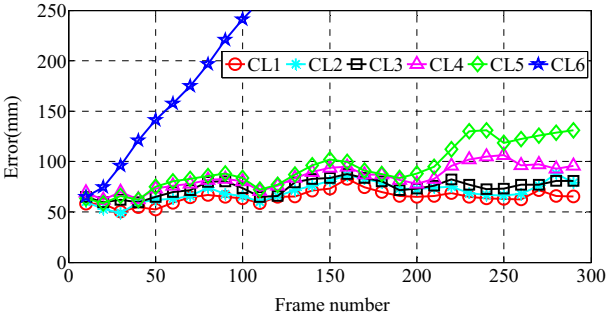

Fig. 17 Tracking errors by our tracker under 6 corruption levels

Table 1 The mean and standard deviation of tracking error (in millimeter) of S2 and S3 calculated over 200 frames

Activity type	Subject 2			Subject 3		
	APF	TGP	Our method	APF	TGP	Our method
Walking	109.41 (50.23)	86.48 (24.53)	66.63 (24.52)	110.42 (52.11)	88.52 (22.54)	68.52 (22.15)
Jogging	119.52 (55.67)	91.53 (24.46)	69.12 (53.63)	112.53 (46.84)	99.89 (22.45)	70.35 (21.29)
Boxing	122.67 (46.42)	90.62 (22.36)	70.42 (21.52)	102.64 (39.63)	84.25 (22.91)	59.52 (24.11)
Gestures	102.33 (44.24)	87.52 (17.65)	67.25 (24.52)	103.25 (34.36)	87.56 (10.85)	50.41 (18.42)
Throw/Catch	106.45 (35.02)	86.52 (17.35)	69.15 (22.24)	118.36 (58.56)	90.63 (25.11)	67.52 (24.01)

8.2.3 Overcoming obstacle occlusion and random noise

To evaluate the capability of our tracking system to handle obstacle occlusion and random noise, several specified areas of test images from HumanEva-II database are replaced by 6 different black rectangular blocks of size 20×30, 25×45, 50×80, 40×60, 60×70 or 70×120. In addition, the salt & pepper noise is added to these occluded images and the noise level is given as the percentage of corrupted pixels. Figure 14 shows some corrupted images with 6 corruption levels (CL). Figure 17 plots the tracking errors by our tracker under 6 different corruption levels (CL1 ~ CL6).

From Fig. 15, we can find that our tracking system has the smallest errors among all algorithms, which demonstrates that our tracker can accurately reconstruct the 3D human pose from the 2D visual observation. Meanwhile, the tracking errors also confirm that our method is more robust than the vector-based methods, such as TGP and APF.

Figure 16 plots the mean and standard deviation of tracking errors of S1 performing 5 kinds of activities. In Table 1, the mean and standard deviation of tracking errors of S2 and S3 are also reported. The boldface denotes the smallest errors. Furthermore, Fig. 18 shows the individual limb errors of S1 performing *Walking*.

8.3 Evaluation: feature and regression algorithm

The accuracy of 3D human pose estimation highly depends on the choice of image feature and regression algorithm. In order to test the impacts of both factors on our

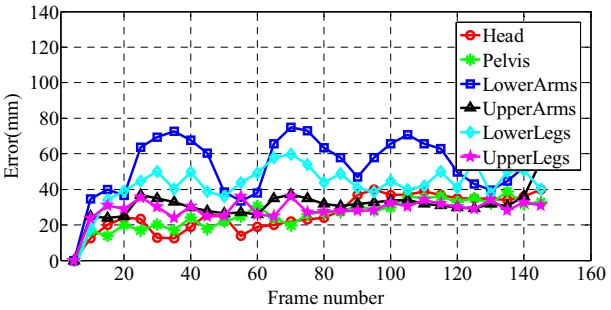

Fig. 18 The individual limb error

tracking tasks, we take both factors into consideration simultaneously in our experiments. We evaluate all human pose regression algorithms on three types of features, i.e., HOG feature, SIFT feature and RAW feature, and implement the experiments combining with the 3-view camera information from cameras C1 ~ C3. We choose three kinds of 3D human pose regression algorithms, i.e., OOSTR, HOSTR [10] and Support Vector Regression (SVR) [31] to recover the 3D human poses. In Table 2, we report the mean (over d joints) root mean square error (RMS) between the estimated joint angle y and the ground-truth y′ [1]. All subjects for 5 kinds of activities (i.e. *Walking*, *Jogging*, *Boxing*, *Gestures* and *Throw/Catch*). The RMS error (in degree) is calculated according to

$$Rms(\mathbf{y}', \mathbf{y}) = \frac{1}{d}\sum_{i=1}^{d} \left|(y_i' - y_i)\text{modulo} \pm 180^{\text{deg}}\right| \tag{60}$$

In Table 2, we can conclude that the performance of OOSTR largely outperforms HOSTR and SVR for three types of features. The results demonstrate that the efficiency of our locally trained models is remarkable. We also find that the performance of the improved HOG feature [17] is better than that of SIFT feature and RAW feature for 3D human pose estimation. Moreover, the performance of SIFT feature is inferior to HOG feature but superior to RAW feature. In other words, the choice of image feature and regression algorithm plays a central role in human pose estimation.

We also report the relative RMS error for each given angle on feature level as illustrated in Fig. 19. In this operation, we choose the *Walking* sequences from HumanEva-I as our test cases. It can be seen that the superiority of HOG feature is remarkable compared with SIFT feature and RAW feature for all regression algorithms.

8.4 Discussions: advantages and limitations

8.4.1 Advantages

This section briefly discusses the advantages of our proposed framework from four aspects:

(1) Our proposed RSTSL handles high-order tensor data and learns multiple sparse tensor subspaces along each tensor mode, which is more effective than the conventional vector-based algorithms that first calculate the vector-based solutions and then convert them to the tensor-based space.
(2) We construct an effective mapping function between LDTF and human pose configuration, which is the first work that learns the human pose regression model in the framework of supervised tensor learning.
(3) We use CP decomposition to capture underlying structure information of an image and reduce loss of information. The method allows multiple projections of LDTF to more than one direction along each tensor mode.
(4) We use the group-sparsity norm based regression that automatically learns the optimal tensor decomposition order, which results in our proposed OOSTR model.

Table 2 The mean RMS error (in degree) over *d* joints. All subjects for *Walking*, *Jogging*, *Boxing*, *Gestures* and *Throw/Catch*

Feature type	Activity type	Subject 1			Subject 2			Subject 3		
		OOSTR	HOSTR	SVR	OOSTR	HOSTR	SVR	OOSTR	HOSTR	SVR
HOG (C1 + C2 + C3)	Walking	5.0413	6.1246	7.0235	4.8610	5.6296	6.8452	5.3309	6.2891	7.2578
	Jogging	3.7012	3.9677	4.1822	4.0432	3.9241	4.7642	4.1131	4.2004	5.0914
	Boxing	4.6737	6.4432	7.2302	5.1467	6.6612	7.5098	4.8203	6.6123	6.9738
	Gestures	5.1024	5.5966	8.0851	4.8053	5.2166	7.8961	4.8014	5.7178	8.2632
	Throw/Catch	4.3144	4.9613	7.3363	4.1441	5.0901	6.9963	4.5285	4.7490	7.0553
SIFT (C1 + C2 + C3)	Walking	5.2122	6.6011	7.4823	5.2092	6.2132	7.4229	6.0622	7.3409	7.7924
	Jogging	4.3932	4.8631	5.1012	4.5521	4.6011	5.0210	4.7322	5.1319	5.7012
	Boxing	5.1324	7.6712	8.1123	5.8397	7.3560	8.2901	5.2097	7.1986	7.5012
	Gestures	6.3011	6.9429	8.8241	6.1412	6.6981	9.8430	5.9081	7.1477	9.6431
	Throw/Catch	5.3324	5.5417	7.9253	5.5091	5.7324	7.7092	5.9671	5.5105	7.5145
RAW (C1 + C2 + C3)	Walking	6.8798	7.6924	7.8773	5.9134	7.4435	7.9094	6.5955	8.0249	7.9672
	Jogging	4.7316	5.4656	5.4544	5.4811	5.5098	6.5672	4.9014	5.7145	6.5614
	Boxing	5.9266	8.2340	9.3664	6.3674	7.9014	9.4120	5.6617	8.1674	8.6081
	Gestures	7.2591	7.4341	9.2647	7.4450	7.6257	9.9434	6.3752	7.7852	10.6015
	Throw/Catch	6.5411	6.3814	8.3431	7.5688	7.0091	9.6001	7.1851	7.2401	8.5785

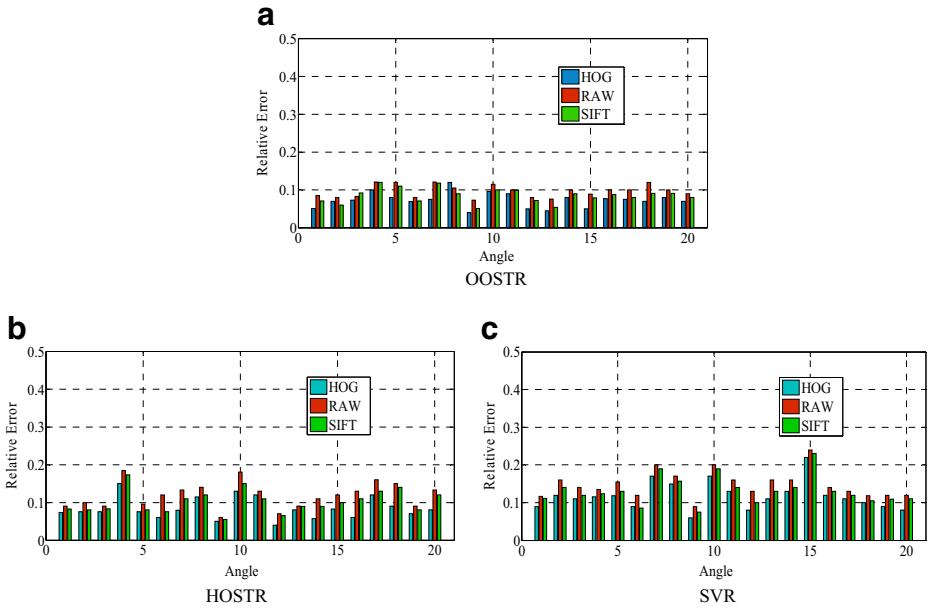

Fig. 19 The superiority comparison between HOG, RAW and SIFT for (**a**) OOSTR, (**b**) HOSTR and (**c**) SVR

8.4.2 Limitations

Additionally, although all the results have demonstrated that our model can achieve stable and accurate 3D human pose tracking, it still has some limitations that will guide our future researches:

(1) The tracking accuracy heavily depends on the training data, namely, our proposed model can only track the human poses that are close to the training samples. In other word, our model is restricted to the human motions provided in the training set. In our future works, we attempt to improve our model to cover a wide range of human motions.
(2) As is known to all, the real body joints are inside the human body while the motion capture markers are placed on the surface of the human body. Thus, the 3D human model is more close to the real human body. Nevertheless, the tracking error computation relies on the Euclidean distances between the estimated joint locations from 3D human model and ground-truth locations from motion capture data. Thus, there exists a systematic bias in our evaluation strategy. Therefore, how to correct this bias is a main problem that we need to handle in our future work.
(3) Both RSTSL and OOSTR models are only used to whole-body 3D human pose representation while the part-based representation is not developed in this paper, which seriously limits the accuracy of human pose estimation.
(4) The 3D human model used in our experiments is based on cylindrical model that cannot be used to simulate some complex 3D human poses, which heavily reduces the range of human motion estimation. In future work, we try to utilize a more effective 3D human model to recover some more complex human poses in the outdoor scenes.

(5) Our model depends on the silhouette features extracted from multi-view image sequence. However, the silhouette extraction is thoroughly difficult especially in the dynamic and complex environments. Therefore, the more robust silhouette extraction approaches and input features need to be considered in our future work.

9 Conclusions

In this paper, we presented a novel method called RSTSL for 3D human pose regression, which extends the generalized tensor subspace learning to a sparse case. Through introducing elastic-net algorithm to optimize the objective function of RSTSL, the most important tensor subspaces for feature extraction can be obtained from the input high-order tensor. Additionally, we presented a novel 3D human pose regression algorithm called OOSTR to estimate the 3D human pose from the low-dimensional tensor feature. Extensive simulations on two publicly available human motion databases (i.e., HumanEva and Brown databases) have verified that both the efficiency and the superiority of our tensor-based methods outperform those of the conventional vector-based ones. In the future, we will improve our model to cover a wide range of human motions and develop a novel mechanism to estimate some new and unseen human motions. Additionally, the more robust silhouette detection approaches and how to correct systematic error will also be considered simultaneously. Finally, a variety of uncontrolled human motion analysis will be implemented in the next work.

Acknowledgments This work is supported by The National Natural Science Foundation of China (Grant: 61202292), Guangdong Province National Science Foundation of China (Grant: 9151064101000037). The authors thank L. Sigal, A. O. Balan and M. J. Black for providing publicly available databases (i.e., Brown and HumanEva databases) for free. The authors would like to thank the anonymous reviewers for their valuable comments that improved our paper.

References

1. Agarwal A, Triggs B (2006) Recovering 3D human pose from monocular images. IEEE Trans Pattern Anal Mach Intell 28(1):44–58
2. Bo LF, Sminchisescu C (2010) Twin Gaussian processes for structured prediction. Int J Comput Vis 87(1):28–52
3. Chretien S, Darses S (2014) Sparse recovery with unknown variance a lasso-type approach. IEEE Trans Inf Theory 60(7):3970–3988
4. Deutscher J, Reid I (2005) Articulated body motion capture by stochastic search. Int J Comput Vis 61(2):185–205
5. Frey BJ, Dueck D (2007) Clustering by passing messages between data points. Science 315(5814):972–976
6. He XF, Cai D (2005) Tensor subspace analysis. In Proc. Advances in Neural Information Processing System (NIPS), pp 499–506
7. Horn RA, Johnson CA (1985) Matrix analysis. Cambridge University Press
8. Hund M, Sturm W, Schreck T, Ullrich T, Keim D, Majnaric L, Holzinger A (2015) Analysis of patient groups and immunization results based on subspace clustering. Proceedings of International Conference on Brain Informatics and Health (BIH), pp 358–368
9. Ionescu C, Carreira J, Sminchisescu C (2014) Iterated second-order label sensitive pooling for 3D human pose estimation. In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp 1661–1668
10. Irene K, Guo WW, Ioannis P (2012) Higher rank support tensor machine for visual recognition. Pattern Recognit 45(12):4192–4203

11. Kolda TG, Bader BW (2009) Tensor decomposition and application. Sian Rev 51(3):455–500
12. Lai ZH, Wong WK, Xu Y, Zhao CR, Sun MM (2014) Sparse alignment for robust tensor learning. IEEE Trans Networks Learn Syst 25(10):1779–1792
13. Lai ZH, Xu Y, Jin Z, Zhang D (2014) Human gait recognition via sparse discriminant projection learning. IEEE Trans on Circuits Syst Video Technol 24(10):1651–1662
14. Lai ZH, Xu Y, Yang J, Tang JH, Zhang D (2013) Sparse tensor discriminant analysis. IEEE Trans Image Process 22(10):3904–3905
15. Lee SC, Ram N (2014) Hierarchical abnormal event detection by real time and semi-real time multi-tasking video surveillance system. Mach Vis Appl 25(1):133–143
16. Lepetit V, Lagger P (2005) Randomized trees for real-time key-point recognition. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp 775–781
17. Li Q, Peng ZL, Lin XM (2015) Unsupervised spectral regression learning for pyramid HOG. J Fiber Bioeng Inf 8(1):117–124
18. Li Y, Sun ZX (2014) Generative tracking of 3D human motion in latent space by sequential clonal selection algorithm. Multimed Tools Appl 69(1):79–109
19. Li Y, Sun ZX, Chen SL (2012) 3D human pose analysis from monocular video by simulated annealed particle swarm optimization. Acta Autom Sin 38(5):732–741
20. Lin WY, Chen YZ, Wu JX, Wang HL, Sheng B, Li HX (2014) A new network-based algorithm for human activity recognition in videos. IEEE Trans Circuits Syst Video Technol 5(24):826–841
21. Ma L, Crawford MM, Yang XQ, Guo Y (2015) Local-manifold-learning-based graph construction for semi-supervised hyperspectral image classification. IEEE Trans Geosci Remote Sens 53(5):2832–2844
22. Panagakis Y, Kotropoulos C (2010) Non-negative multi-linear principal component analysis of auditory temporal modulations for music gene classification. IEEE Trans Acoust Speech Signal Process 18(3):576–588
23. Qin ZW, Scheinberg K (2014) Efficient block-coordinate descent algorithms for the group lasso. Math Program Comput 5(2):143–169
24. Raskin L, Rudzsky M, Rivlin E (2011) Dimensionality reduction using a Gaussian process annealed particle filter for tracking and classification of articulated body motions. Comput Vis Image Und 115(4):503–519
25. Rosales R, Sclaroff S (2006) Combining generative and discriminative models in a framework for articulated pose estimation. Int J Comput Vis 67(3):251–276
26. Sigal L, Bhatia S, Roth S, Black MJ, Isard M (2004) Tracking loose-limbed people. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp 1421–1428
27. Sigal L, Black MJ (2010) HumanEva: synchronized video and motion capture dataset for evaluation of articulated human motion. Int J Comput Vis 87(1):4–27
28. Steve RG (1997) Support vector machines for classification and regression. ISIS Technical Report, University of Southampton
29. Tan X, Wu F, Li X, Tang SL, Lu WM (2015) Structured visual feature learning for classification via supervised probabilistic tensor factorization. IEEE Trans Multimed 17(5):660–673
30. Tao DC, Li XL, Wu XD, Maybank SJ (2007) General tensor discriminant analysis and Gabor feature for gait recognition. IEEE Trans Pattern Anal Mach Intell 29(10):1700–1715
31. Vapnik VN (1995) The nature of statistical learning theory. Springer, New York, pp 219–224
32. Wang JM, Fleet DJ, Hertzmann A (2008) Guassian process dynamical models for human motions. IEEE Trans Pattern Anal Mach Intell 30(2):283–298
33. Wu X (2010) Tensor-based projection using ridge regression and its application to each classification. IET Image Proc 4(6):486–493
34. Yang SY, Jin PL, Li B, Yang LX, Xu WH, Jiao LC (2014) Semi-supervised dual-geometric subspace projection for dimensionality reduction of hyperspectral image data. IEEE Trans Geosci Remote Sensi 52(6): 3587–3593
35. Yao A, Gall J, Luc VG, Urtasun R (2011) Leaning probabilistic non-linear latent variable models for tracking complex activities. In Proc. Advances in Neural Information Processing System (NIPS), pp 1359–1367
36. Zhang Z, Yang X, Oseledets IV, Karniadakis GE, Daniel L (2015) Enabling high-dimensional hierarchical uncertainty quantification by ANOVA and tensor-train decomposition. IEEE Trans Comput Aided Des of Integr Circuits Syst 34(1):63–76
37. Zhao X, Fu Y, Ning HZ, Liu YC, Huang TS (2010) Human pose regression through multi-view visual fusion. IEEE Trans Circuits Syst Video Technol 20(7):957–966
38. Zhu R, Yuan JS, Meng JJ, Zhang ZY (2013) Robust part-based hand gesture recognition using kinect sensor. IEEE Trans Multimed 15(5):1110–1120
39. Zolfaghari M, Jourabloo A, Gozlou SG, Pedrood B, Manzuri-Shalmani MT (2014) 3D human pose estimation from image using couple sparse coding. Mach Vis Appl 25(6):1489–1499

40. Zou H, Hastie T (2005) Regression shrinkage and selection via the elastic net, with applications to microarrays. J Royal Stat Soc B 67(1):301–320
41. Zou H, Hastie T, Tibshirani B (2006) Sparse principle component analysis. J Comput Graph Stat 15(2):265–286

Jialin Yu received the B.S. degree in Electronic Information Science and Technology from the College of Science of Guizhou University, Guizhou, China, in 2013. From September 2014, he is pursuing to the Ph.D. degree at the College of Electronic and Information Engineering of South China University of Technology. His current research interests include computer vision, pattern recognition and machine learning.

Jifeng Sun received the B.S. degree in Machine Building & Automation from Tsinghua University, Beijing, China, in 1983, the M.S. degree in Precision Instruments and Testing from Tsinghua University, Beijing, China, in 1986, and the Ph.D. degree in Electrical Engineering from Kyushu University, Fukuoka, Japan, in 1995. From 2005 to 2009, he was a Chief Professor of Computer Vision and Intelligent Information Processing Laboratory. He is currently a professor and Ph.D. supervised of the Department of Electronic and Information Engineering. His current research interests include computer vision, machine learning and self-organizing communication networks.