Mining Spatial-Temporal Patterns and Structural Sparsity for Human Motion Data Denoising

Yinfu Feng, Mingming Ji, Jun Xiao, Xiaosong Yang, Jian J. Zhang, Yueting Zhuang, and Xuelong Li, *Fellow, IEEE*

***Abstract*—Motion capture is an important technique with a wide range of applications in areas such as computer vision, computer animation, film production, and medical rehabilitation. Even with the professional motion capture systems, the acquired raw data mostly contain inevitable noises and outliers. To denoise the data, numerous methods have been developed, while this problem still remains a challenge due to the high complexity of human motion and the diversity of real-life situations. In this paper, we propose a data-driven-based robust human motion denoising approach by mining the spatial-temporal patterns and the structural sparsity embedded in motion data. We first replace the regularly used entire pose model with a much fine-grained partlet model as feature representation to exploit the abundant local body part posture and movement similarities. Then, a robust dictionary learning algorithm is proposed to learn multiple compact and representative motion dictionaries from the training data in parallel. Finally, we reformulate the human motion denoising problem as a robust structured sparse coding problem in which both the noise distribution information and the temporal smoothness property of human motion have been jointly taken into account. Compared with several state-of-the-art motion denoising methods on both the synthetic and real noisy motion data, our method consistently yields better performance than its counterparts. The outputs of our approach are much more stable than that of the others. In addition, it is much easier to setup the training dataset of our method than that of the other data-driven-based methods.**

***Index Terms*—Human motion denoising, $\ell_{2,p}$-norm, Microsoft Kinect, motion capture data, robust dictionary learning, robust structured sparse coding.**

Manuscript received September 2, 2014; revised November 30, 2014; accepted December 3, 2014. Date of publication December 30, 2014; date of current version November 13, 2015. This work was supported in part by the National Key Basic Research Program of China under Grant 2012CB316400, in part by the National High Technology Research and Development Program under Grant 2012AA011502, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY13F020001, in part by the Fundamental Research Funds for the Central Universities under Grant 2014FZA5013, in part by Zhejiang Province Public Technology Applied Research Projects under Grant 2014C33090, and in part by the grant of the Sino-U.K. Higher Education Research Partnership for Ph.D. student's project funded by the Department of Business, Innovation, and Skills of the British Government and the Ministry of Education of China. This paper was recommended by Associate Editor L. Shao.

Y. Feng, M. Ji, J. Xiao, and Y. Zhuang are with the College of Computer Science, Zhejiang University, Hangzhou 310027, China (e-mail: junx@cs.zju.edu.cn).

X. Yang and J. J. Zhang are with the National Centre for Computer Animation, Bournemouth University, Poole BH12 5BB, U.K.

X. Li is with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2381659

I. Introduction

MOTION capture also known as motion tracking is a prevalent technique to record the movement of objects or people for immediate or delayed motion analysis and reuse [1]–[3]. With the rapid development of motion capture techniques and systems in the past two decades, more and more motion data are available. Motion data have been used in a wide variety of fields, such as computer vision [4]–[7], computer animation [8], [9], movie production, virtual reality, and medical rehabilitation. In the movie industry, the great success of recent science fiction action films represented by *Avatar*, *The Avengers*, *Transformers: Age of Extinction*, and *Dawn of the Planet of the Apes*, wherein high quality motion data have been extensively adopted for generating character animation, facial animation, and special effects, demonstrates the important application values of motion capture techniques and data.

To acquire motion data, the available techniques can either be inertial, mechanical, magnetic, optical, and depth-based motion capture [10]–[13]. Among these existing techniques, the optical-based motion capture technique has attracted much attention since the performer is freer to move and the captured motion data are more accurate. However, even with the professional optical-based motion capture systems such as motion analysis system and Vicon, the captured raw data often contain inevitable noises and outliers [14]–[19]. For example, when some markers are occluded by the human body or objects, they become invisible to the cameras, which leads to missing data. But whatever data prediction methods were used to fill the missing data, it may bring in a certain percentage of noise. If two markers are mislabeled when the tacking algorithm confuses the trajectory of one marker with that of the other in some cases, the captured raw data will contain serious errors, which also can be regarded as bad noises or outliers. Two real examples are given in Fig. 1. From Fig. 1(b), we can see that the positions where the missing data appeared to exhibit a strong structural distribution property. In Fig. 1(a), the noisy joints distort human poses and make the whole motion become unnatural and unsmooth.

The processing of human motion capture is usually both expensive and time consuming. Thus, it is essential to reuse the captured motion data. To achieve this goal, the first task is to refine the captured raw motion data by removing the noise and outliers from them. In practice, most commercial motion capture systems provide some post-processing tools for cleaning motion data, i.e., filling missing values and removing noises.

2168-2267 © 2014 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/ redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

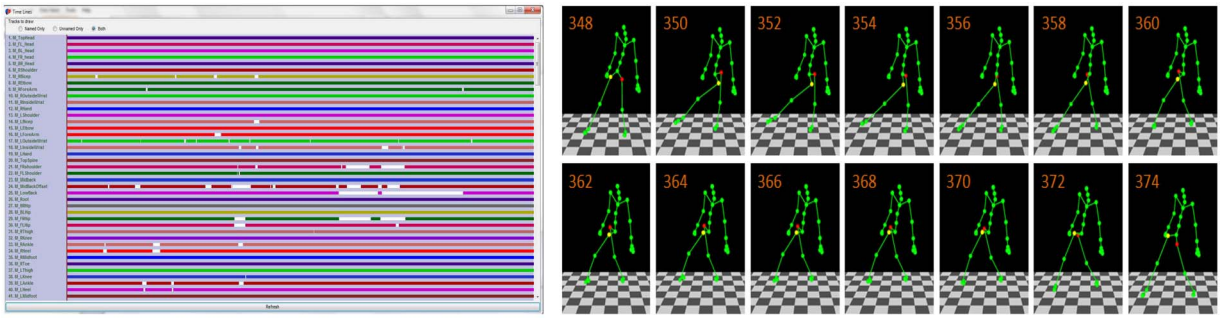


Fig. 1. Two real missing and noisy motion capture data sequences. (a) Time lines of all markers in a jump motion sequence, which is captured by ourselves using a MotionAnalysis Raptor-E Digital RealTime System produced by Motion Analysis Corporation. The white blank spaces within the color lines represent that the corresponding markers are lost at that frame. (b) Several poses from one motion sequence (i.e., 83_68.*amc*) are selected from the public Carnegie Mellon University (CMU) human motion capture dataset [20]. The performed human action in this motion sequence is medium sidestep to right. To be convenient for identification, the two noisy joints are marked using red and yellow colors separately in each motion frame.

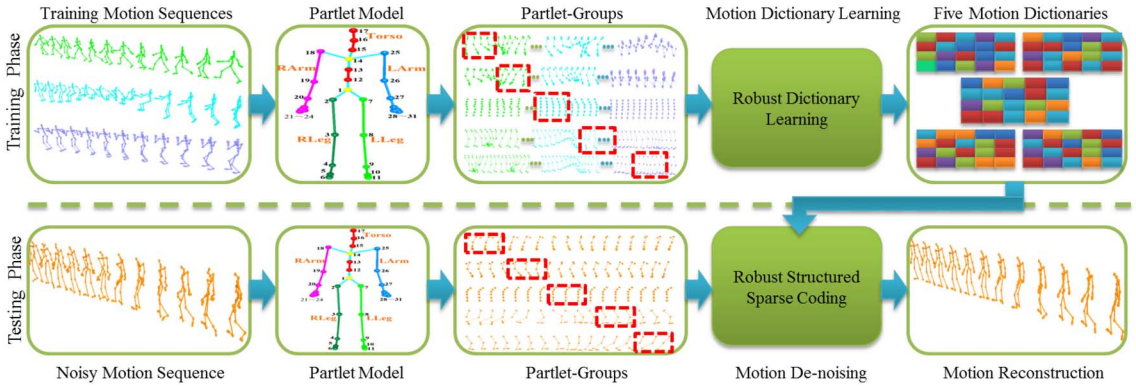


Fig. 2. Flowchart of our proposed human motion data denoising approach. For the input motion sequences, we first divide each human pose into five partitions that are termed as partlets. These partlets are then grouped using a lagged window moving through all of the motion sequences to generate multiple partlet-groups. In the training phase, we use these partlet-groups to learn five motion dictionaries via a robust dictionary learning algorithm and adopt them to remove the noise and outliers from noisy partlet-groups in the testing phase (i.e., the denoising phase) by optimizing a robust structured sparse coding problem. Finally, we reorganize the filtered partlet-groups to reconstruct the filtered clean motion sequences.

However, it requires the user to inspect each motion sequence frame-by-frame and correct the noisy and missing markers one-by-one, making it time-consuming and error-prone [17]. Additionally, the underlying denoising/smoothing methods of these tools are often based on interpretation, which means that they are efficient only for dealing with certain simple and short-term noise cases. They would fail to handle complex and long-term noisy cases. Furthermore, the spatio-temporal patterns, which are embedded in motion data, play an important role in characterizing human motion. However, they have been ignored by these methods, and consequently result in distorted and unrealistic motions. In addition, the emergence of low-cost depth sensors (e.g., Microsoft Kinect) that can acquire a depth stream with acceptable accuracy provides new opportunities for accessible motion capture in recent years. With the depth stream, it is possible to estimate human motion in real-time [13], [21], [22], although the acquired depth data are quite noisy and many pixels in the image may have no depth suffer from multiple reflections, transparent objects or scattering problems in certain surfaces (such as human tissue and hair) [12]. The motion data derived from the depth stream are more noise than that from the traditional motion capture techniques. Researchers still have a long uphill journey in improving the quality of the data.

To denoise the imperfect motion data, a lot of motion denoising methods have been proposed in the literature. However, some intrinsic shortcomings hinder them from being widely applied in real-world applications [17], [19]. For example, the structural relationship between different human joints and the spatio-temporal patterns embedded in human motion [17], [23]–[26] have not been well exploited in most existing methods. In order to overcome these problems, in this paper, we propose a novel data-driven-based robust human motion denoising approach deriving from dictionary learning and sparse coding theories. The main ideas of our method are twofold: to sparsely select the most correlated subset of motion bases for clean motion reconstruction and to take into account the distribution information of noises and outliers in motion data in deriving our objective functions. The flowchart of our approach is illustrated in Fig. 2. And, the major contributions of the proposed approach include the following.

1) A fine-grained human pose representation model named partlet model is proposed in this paper. Using the entire human pose model as feature representation is a little

coarser, and noisy data on one part of human body will inevitably influence the clean data at other parts. To avoid this problem, we divide each human pose into five parts named partlets to obtain a much fine-grained representation. One potential benefit is that these five parts may be processed in parallel making it much fast to compute. As shown in our experiments, using such a new representation does not only reduces the entire data processing time, but also improves the performance of our approach. Another significant benefit is that the out-of-sample problem of data-driven-based methods can be mitigated by exploiting the abundant local body part posture and movement similarities embedded in different human motion sequences if we use the new representation. It can handle the new-come motion sequence as long as similar local body part postures and movements have been collected in the training dataset. Fundamentally, we relax the requirements (i.e., similar motion sequences in the training dataset) for overcoming the out-of-sample problem.

2) By simply decomposing the motion noise into dense Gaussian noise and sparse outliers, a robust dictionary learning algorithm is proposed to learn multiple motion dictionaries, which contain the spatio-temporal patterns of human motion. Our approach can robustly learn motion dictionaries from both clean and noisy training dataset. Thus, it is easier to collect motion sequences to setup the training dataset for our approach than that of the other data-driven-based methods [17].
3) By utilizing the sparse sample selection ability of the ℓ_1-norm [27]–[29], we convert the conventional human motion denoising problem into a general ℓ_1-minimization framework. In contrast with [17], our method can automatically select the most correlated subset of motion bases from motion dictionaries for clean motion reconstruction. As a result, we do not need to specifically choose the training dataset, and our method can be more easily applied to real-world applications.
4) We explicitly take both the noise structure information and the motion smoothness constraint into account in a joint framework. We enforce a $\ell_{2,p}$-norm penalty on the noise term to exploit the structure information of noise. And, the $\ell_{2,p}$-norm provides more choices of $p \in (0, 2]$ to fit variety of jointly sparse structure of noise. Meanwhile, we incorporate a smooth graph constraint on the sparse representation coefficients matrix in our objective function to make sure the refined human motion as smooth as possible.

The structure of this paper is organized as follows. We first introduce some related work in Section II, and then provide the details of our proposed approach in Section III. The experimental analysis and the conclusion are finally given in Sections IV and V, respectively.

II. Related Work

The goal of human motion denoising is not only to remove noises and outliers from motion data but also to preserve both the embedded spatio-temporal patterns of human motion and the human body structural constraints (e.g., the bone-length constraints). Due to the high complexity of human motion, human motion denoising is a very challenging task, and a lot of research effort has been expended on this topic.

In some earlier studies, the classic signal denoising methods like Gaussian low-pass and wavelet transformation are adopted to filter motion data [16], [17]. For example, a B-spline wavelet-based method is proposed to remove the impulsive noise embedded in noisy rigid body motion data [16]. The biggest advantage of these methods is that they are very fast and require little computational cost. However, they process each feature dimension of motion data independently, ignoring the structural relationship between different human markers/joints and the spatio-temporal patterns embedded in motion data.

Another way is to apply linear time-invariant filters to denoise noisy motion data [14], [30]. Lee and Shi [14] formulated filtering nonlinear motion orientation data into a linear time-invariant filtering framework by transforming the orientation data into a vector space and then transforming the results back to the orientation space after filtering. Yamane and Nakamura [30] presented a dynamic filter that converts a physically inconsistent motion into a consistent one. These methods unfortunately also suffer from the same shortcomings as the signal denoising methods.

As an improvement, dynamic system-based methods represented by Kalman filter and linear dynamic system (LDS) are used to discover the hidden variables and learn their dynamics [31], [32]. Shin *et al.* [31] used a Kalman filter scheme that addresses motion capture noise issues in real-time computer puppetry situation. Li *et al.* [32] formulated motion completion problem as a constrained optimization problem with the framework of LDS. Since dynamic system-based methods predict the current state relying on the past information, the filtered motion exhibit a little of time delay that cannot meet the real-time requirement for some applications [19].

With the explosive growth of available motion capture data, data-driven-based methods [17], [26] have attracted much attention. Lou and Chai [17] proposed an example-based human motion denoising method that first applies multichannel singular spectrum analysis to learn a series of filter bases, which hold spatio-temporal patterns embedded in the precaptured clean motion data, and then use them along with robust statistics techniques to filter noisy motion data. Their method received encouraging results both on the simulated and real motion data. However, the data-driven-based methods suffer from three fundamental problems: 1) their performance relies heavily on the clean training dataset, while both the training and the testing data may contain noises and outliers in practice; 2) the training datasets in these methods must be selected carefully and they only contain the motion sequences that come from the same type of human action as the testing/noisy motion sequences; and 3) they are unable to handle the new-come motion sequence when there are no similar motion sequences in the training dataset. The last problem is also called as the out-of-sample problem. Besides, we have to point

out that only a subset of filter bases are remained and used to filter the noisy motion in [17], so their method is unable to preserve all motion details in the original motion sequences. Indeed, the motion dictionary matrix used in [17] is not a full-rank matrix, thus the filter bases do not span the whole motion space. In contrast, five overcomplete motion dictionaries are learned and used in this paper, making it possible to hold all motion details.

Recently, Lai *et al.* [18] reformulated the human motion completion and denoising problems into a low-rank matrix optimization framework based on the observation that motion matrices are mostly approximately low-rank. The greatest advantage of their method is that their method can process each input motion sequence independently without the support of training dataset. However, the low-rank matrix completion theory would not be applicable when many data entries are badly corrupted. Moreover, the user has to guess the standard deviation of noise in their work, which is difficult in practice.

Arguably, the human motion data denoising problem is still an open problem. The great success of data-driven-based methods [5], [7], [25], [29], [33], [34] in computer vision and machine learning encourages us to propose a novel yet robust human motion data denoising approach to overcome the existing issues.

III. Methodology

In this section, we present a data-driven-based robust human motion denoising approach deriving from dictionary learning and sparse coding theories as shown in Fig. 2.

A. Notations and Definitions

In this paper, capital letters, e.g., X, represent matrices or sets, while lower case letters, e.g., x, represent vectors or scale values. X_{ij} is the (i, j)th entry of X, and $X_{i\cdot}$ denotes the ith row of X, while $X_{\cdot i}$ denotes the ith column of X. Similarly, x_i is the ith element of vector x. For an $m \times n$ matrix $X = [x_{ij}]$, let $\text{vec}(X) = (x_{11}, \ldots, x_{m1}, x_{12}, \ldots, x_{mn})^T$ be the $mn \times 1$ vector. Here $\text{vec}(\cdot)$ is defined as the vectorization operation that reshapes a matrix into a vector by stacking all columns one-by-one. $X \otimes Y = [x_{ij}Y]$ represents the Kronecker product of X and Y. I_c represents the $c \times c$ identity matrix. For any vector $x \in \mathcal{R}^d$, several useful vector norms are defined

$$\|x\|_0 = \sum_{x_i \neq 0} \|x_i\|^0, \quad \|x\|_1 = \sum_{i=1} |x_i|, \quad \|x\|_p^p = \sum_{i=1} |x_i|^p. \quad (1)$$

Similarly, for any matrix $X \in R^{m \times n}$, the squared Forbenius norm (i.e., the ℓ_2-norm), the ℓ_1-norm, the $\ell_{2,1}$-norm, and the $\ell_{2,p}$-norm can be defined as

$$\|X\|_F^2 = \sum_{ij} X_{ij}^2, \quad \|X\|_{2,1} = \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{n} X_{ij}^2}$$

$$\|X\|_1 = \sum_{ij} |X_{ij}|, \quad \|X\|_{2,p} = \left(\sum_{i=1}^{m} \left(\sum_{j=1}^{n} |X_{ij}|^2 \right)^{\frac{p}{2}} \right)^{\frac{1}{p}}. \quad (2)$$

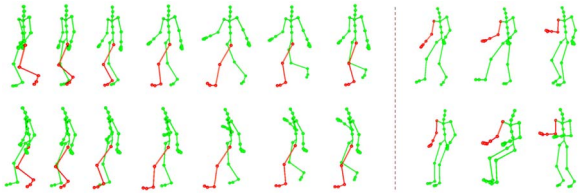
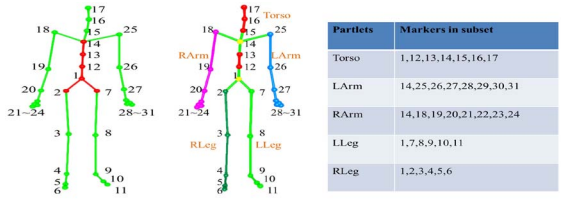


Fig. 3. Similar movements and postures of human body parts in motion sequences. Left: walk and a basketball motion sequences with similar leg movements. Right: six poses with similar hand postures selecting from basketball and jump motion sequences separately.



Partlets	Markers in subset
Torso	1,12,13,14,15,16,17
LArm	14,25,26,27,28,29,30,31
RArm	14,18,19,20,21,22,23,24
LLeg	1,7,8,9,10,11
RLeg	1,2,3,4,5,6

(a) (b)

Fig. 4. CMU (a) pose model and our (b) partlet model. The rigid part is marked with red color in the left figure. The markers 1, 2, 7, and 14 are the root, the right and left femur markers, and the upper neck marker, respectively.

B. Data Preprocessing

1) Coordinate Normalization: Since the raw motion data are recorded under the real-world global coordinate system, visual-similar human poses are possible to be of dramatically numerical diversity due to the reasons of pose translation and rotation. Meanwhile, different motion sequences frequently contain an amount of similar body part postures and movements under the local coordinate system as shown in Fig. 3. In view of this, we hope to remove the effect of pose translation and rotation in pose representation while to exploit the available local body part similarities for human motion denoising. We also notice that the torso of a human body as marked in Fig. 4(a) is usually a stable rigid part. Referring to the human torso, it becomes simple yet efficient to calculate both the translation and rotation matrices for each human pose. It allows us to remove the translation and rotation of the input human poses, and is easy to set them in a local coordinate system offering an invariant pose representation. To this end, we first normalize each human pose and translate it to make its root marker in the origin point of the local coordinate system. Then, we align the local pose by rotating it so that the plane consisting of three markers, i.e., the left femur, the right femur, and the upper neck, parallels the XY-plane. And, the ray that passes through the middle point between the left and right femur markers and the upper neck marker also parallels the y-axis, and directs to the positive direction of the y-axis. In order to take the effect of noise into account, we adopt the iterative closest point algorithm [35] to obtain the translation and rotation matrices. We record all of these transformation information into a matrix $M = M_r \times M_t$, where M_r is the rotation matrix and M_t is the translation matrix, so all of these operations can be done in reverse. In other words, we can translate these local poses into global poses after human motion denoising.

2) Partlet Generation: Suppose a normalized local human motion sequence comprises n poses and each pose contains l markers, we denote it as $X = [X_{\cdot 1}, \ldots, X_{\cdot n}]$, where $X_{\cdot t} = [x_{1t}, y_{1t}, z_{1t}, \ldots, x_{lt}, y_{lt}, z_{lt}]^T$ represents the tth pose and (x_{it}, y_{it}, z_{it}) is the coordinate of the ith marker in this pose.

Because all l markers are included in pose representation (i.e., $X_{\cdot t}$), we call such a representation as the entire pose model representation, which has been frequently used in [18], [19], [32], [36], and [37]. However, this pose representation has two shortcomings: 1) the badly noisy body parts will inevitably affect the clean body parts and 2) it is too coarser to exploit the abundant similar local body part postures and movements. To improve the representation, we divide each pose into five parts termed as the partlets according to the human anatomy in order to obtain a more fine-grained representation. The five partlets are torso (contains head), left arm, right arm, left leg, and right leg. Each of them is a set of markers as shown in Fig. 4(b). In order to make the position of the joint markers like markers 1 and 14 in Fig. 4 stable, we assign them to multiple partlets as shown in the table of Fig. 4(b). For each partlet, one submatrix is derived from X and we denote the ith partlet as $X^i = [X^i_{\cdot 1}, \ldots, X^i_{\cdot n}] \in \mathcal{R}^{d_i \times n}, i = 1, \ldots, 5$, wherein $X^i_{\cdot t}$ just includes a subset of markers of $X_{\cdot t}$ and d_i equals to three timing the number of contained markers in the ith partlet. With this representation, we can speed up human motion denoising via processing these five partlets in parallel. An incidental benefit is that the out-of-sample problem can be mitigated via exploiting the abundant similar local body part postures and movements embedded in different human motion sequences. It is able to deal with the new-come motion sequence as long as similar local body part postures and movements are collected in the training dataset. In other words, we relax the requirements (i.e., similar motion sequences are collected in the training dataset) for overcoming the out-of-sample problem.

3) Partlet Grouping: If we process each human pose one-by-one, the embedded spatial-temporal motion patterns will be ignored. In other words, it would be much better to process a short clip of motion than a single pose each time. In light of this, we use a lagged window with the length of m-frames moving across the entire motion sequence as shown in Fig. 2 and group all of the partlets in a same window into a group. The above obtained partlets in X^i are reorganized into $n - m + 1$ overlapped groups. We then reshape each group into a vector $\mathbf{g}^i_j = \mathrm{vec}([X^i_{\cdot j}, X^i_{\cdot j+1}, \ldots, X^i_{\cdot j+m-1}]) \in \mathcal{R}^{(d_i \times m) \times 1}$, where $j = 1, \ldots, n - m + 1$. So, we totally obtain five partlet-group motion matrices, i.e., $Y^i = [\mathbf{g}^i_1, \ldots, \mathbf{g}^i_s], s = n - m + 1$, from $X^i, i = 1, \ldots, 5$, by partlet grouping operation.

C. Motion Dictionary Construction Via Robust Dictionary Learning

Human motion data contain some intrinsic spatio-temporal motion patterns, and it is helpful to reveal and exploit these patterns for guiding human motion denoising. Different from [17], we resort to the theory of dictionary learning [4], [25], [38], [39] to adaptively infer five motion dictionaries that preserve the spatio-temporal patterns of human motion from a training dataset.

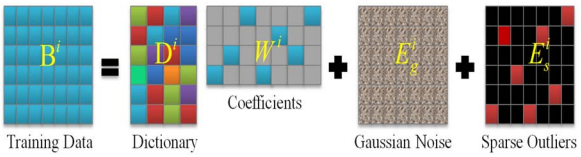


Fig. 5. Sketch map of robust dictionary learning algorithm.

Specifically, collecting as many precaptured complete motion sequences as possible to construct a training dataset is applied as the first step. Once such a training dataset is set up, we can use the training motion sequences to construct five partlet-group motion matrices, i.e., $B^i = [g^i_1, \ldots, g^i_N] \in \mathcal{R}^{(d_i \times m) \times N}, i = 1, \ldots, 5$, wherein N is the total number of partlet group, via the partlet generation and grouping operations. The next step is to solve the following optimization problem:

$$\begin{aligned} &\min_{D^i, W^i} \left\| B^i - D^i W^i \right\|_F^2 \\ &\text{s.t.} \quad W^i = \left[W^i_{\cdot 1}, \ldots, W^i_{\cdot N} \right], \left\| W^i_{\cdot j} \right\|_0 \le t_s, 1 \le j \le N; \\ &\qquad D^i = \left[D^i_{\cdot 1}, \ldots, D^i_{\cdot K_i} \right], \left\| D^i_{\cdot j} \right\|_2 \le 1, 1 \le j \le K_i \end{aligned} \quad (3)$$

to search the best possible motion dictionary (i.e., $D^i \in \mathcal{R}^{(d_i \times m) \times K_i}$) for the sparse representation of the training partlet-group data B^i. In (3), W^i is the sparse representation coefficient matrix, t_s is the target sparsity and $\| \cdot \|_0$ is the ℓ_0 pseudo-norm as defined in (1). In each motion dictionary matrix, e.g., $D^i, i = 1, \ldots, 5$, its columns are the desired motion bases which preserve the embedded spatio-temporal patterns of human motion. For simplicity, we assume hereon that the columns of D^i are normalized to unit ℓ_2-length. Equation (3) is actually a nonconvex problem with respect to D^i and W^i, while there exist several efficient dictionary learning methods such as the classic MOD [40], K-SVD [41], and their variants [42] can be used to solve it.

However, (3) is a least square error function which is well-known to be unstable with respect to noises and outliers [43]. Suppose B^i is contaminated by a few outliers with large errors, these outliers are easy to dominate (3) because of the squared errors. Indeed, (3) is only optimal when B^i is contaminated by the independent and identically distributed (i.i.d.) Gaussian noise. In practice, we often have to face the gross noise condition when the motion data have been badly corrupted by the noise and outliers. So, it is beneficial to enhance the robustness of (3).

1) Objective Function: In the specific area of human motion capture, the acquired human motion data usually contain only a few of outliers after post processing. Thus, the outliers in B^i are rare comparing with the dense slight Gaussian noise in it. In order to distinguish these two kinds of noise, we decompose the whole noise E^i in B^i into two parts: 1) one is the dense Gaussian noise E^i_g and 2) the other one is the sparse outliers E^i_s, so we have $E^i = E^i_g + E^i_s$. The key idea of our proposed robust dictionary learning algorithm is shown in Fig. 5.

Since (3) is optimal to the Gaussian noise, we omit E_g^i and propose a robust dictionary learning algorithm by minimizing the following objective function [44], [45]:

$$\begin{aligned}&\min_{D^i,W^i,E_s^i} \left\|B^i - D^iW^i - E_s^i\right\|_F^2 + \lambda \left\|E_s^i\right\|_1 \\ &\text{s.t.} \quad W^i = \left[W_{\cdot 1}^i, \ldots, W_{\cdot N}^i\right], \left\|W_{\cdot j}^i\right\|_0 \le t_s, 1 \le j \le N \\ &\qquad D^i = \left[D_{\cdot 1}^i, \ldots, D_{\cdot K_i}^i\right], \left\|D_{\cdot j}^i\right\|_2 \le 1, 1 \le j \le K_i. \quad (4)\end{aligned}$$

Due to the ℓ_0 pseudo-norm in (4), it is hard to solve the optimization problem. As customary, we relax it by minimizing its ℓ_1 surrogate and reformulate the objective function as follows:

$$\begin{aligned}&\min_{D^i,W^i,E_s^i} \left\|B^i - \left[D^i, I\right]\begin{bmatrix} W^i \\ E_s^i \end{bmatrix}\right\|_F^2 + \lambda \left\|E_s^i\right\|_1 + \beta \left\|W^i\right\|_1 \\ &\text{s.t.} \quad D^i = \left[D_{\cdot 1}^i, \ldots, D_{\cdot K_i}^i\right], \left\|D_{\cdot j}^i\right\|_2 \le 1, 1 \le j \le K_i \quad (5)\end{aligned}$$

where I is a $(d_i \times m) \times (d_i \times m)$ identity matrix. Compared the improved (5) with the original (3), since we explicitly take the effect of noises and outliers into account in deriving the proposed robust dictionary learning algorithm, the algorithm becomes robust against with the motion data noises and outliers.

2) Optimization Method: Since (5) is actually a nonconvex problem with respect to D^i, W^i and E_s^i jointly, it is difficult to find the global minimum. However, (5) is convex with the three variables separately. In the following, we use a coordinate descent scheme to optimize the three variables alternatively.

For the fixed W^i and E_s^i, (5) is equivalent to

$$\begin{aligned}&\min_{D^i} \left\|B^i - D^iW^i - E_s^i\right\|_F^2 \\ &\text{s.t.} \quad D^i = \left[D_{\cdot 1}^i, \ldots, D_{\cdot K_i}^i\right], \left\|D_{\cdot j}^i\right\|_2 \le 1, 1 \le j \le K_i \quad (6)\end{aligned}$$

which is a least squares problem with quadratic constraints. This constrained optimization problem can be solved using gradient descent with iterative projection, while it also can be much more efficiently solved using a Lagrange dual [46].

Once the motion dictionary D^i is updated, we optimize W^i and E_s^i in (5)

$$\min_{W^i,E_s^i} \left\|B^i - \left[D^i, I\right]\begin{bmatrix} W^i \\ E_s^i \end{bmatrix}\right\|_F^2 + \lambda \left\|E_s^i\right\|_1 + \beta \left\|W^i\right\|_1. \quad (7)$$

Equation (7) becomes a classic ℓ_1 minimization problem and it can be solved using the orthogonal matching pursuit [47], basis pursuit [48], FOCUSS [49] and fixed-point continuation (FPC) algorithm [50], and so on.

Therefore, we iteratively update D^i, W^i, and E_s^i until the convergence is achieved, leading to a local optimum solution of these three variables. Therefore, five motion dictionaries $D^i, i = 1, \ldots, 5$ can be get in the training phase via the proposed robust dictionary learning algorithm.

D. Human Motion Denoising Via Robust Structured Sparse Coding

Similarly, using the partlet generation and grouping operations, we can generate five partlet-group motion matrices, which are denoted as $Y_t^i, i = 1, \ldots, 5$, for an input noisy motion sequence in the testing/denoising phase. For a number of reasons, the partlet-groups (i.e., g_j^i) in Y_t^i contain noises and outliers. A slight difference between B^i and Y_t^i is that the former contains multiple partlet-groups from different motion sequences while the later one just contains multiple partlet-groups from only one motion sequence. This permits us to exploit the noise distribution information within a single motion sequence for human motion denoising. In order to remove the noises and outliers, we convert the conventional human motion de-noising problem into a general ℓ_1-minimization framework, which can automatically select a most correlated subset of motion bases from the motion dictionary for clean motion reconstruction.

1) Objective Function: To construct and select sparse motion bases, we utilize the ℓ_1-norm, which is of the sparse sample selection ability [27], and reformulate the human motion de-noising problem into a ℓ_1-minimization framework

$$\min_{\Psi^i} \text{loss}\left(Y_t^i, D^i\Psi^i\right) + \mathcal{G}\left(\Psi^i\right) + \|\Psi^i\|_1. \quad (8)$$

The added ℓ_1-norm penalty on the coefficient matrix Ψ^i cannot ensure the filtered human motion to be smooth, so we incorporate a smooth graph constraint on Ψ^i, i.e., $\mathcal{G}(\Psi^i)$. In fact, the graph-based models have been applied in many applications, such as visual concept recognition [19], [43], [51], [52] and photo retargeting and cropping [53]. Here the smooth graph denotes as $G^i = \{Y_t^i, S^i\}$, where the data samples in Y_t^i as graph vertices and S^i is the graph weight matrix, whose element S_{ab}^i reflects the visual similarity between $\mathbf{g}_a^i$ and $\mathbf{g}_b^i$ in Y_t^i. There exist two popular ways to assign the graph weight matrix: 1) one is the k-nearest-neighbor method and 2) the other is the ϵ-ball-based method. To reduce the number of parameters, we adopt the former one and define S^i as below

$$S_{ab}^i = \begin{cases} 1 \text{ if } \mathbf{g}_a^i \in \mathcal{N}_k\left(\mathbf{g}_b^i\right) \text{ and vice versa} \\ 0 \text{ otherwise} \end{cases}$$

where $\mathcal{N}_k(\mathbf{g}_b^i)$ is the k-nearest neighborhood set of $\mathbf{g}_b^i$. To enhance the robustness, we adopt ℓ_1-norm distance, i.e., $\|\mathbf{g}_a^i - \mathbf{g}_b^i\|_1$, as the measurement to find the k-nearest neighborhoods. We empirically set k to 5 in our experiments. Because of the smoothness of human motion, the temporal neighbors of $\mathbf{g}_a^i$ are of a highly probability to be in the k-nearest neighborhood set. To enforce the recovered human motion to be smooth, $\mathcal{G}(\Psi^i)$ is defined as

$$\mathcal{G}\left(\Psi^i\right) = \frac{1}{2}\sum_{a,b=1}^{s}\left(\Psi_{\cdot a}^i - \Psi_{\cdot b}^i\right)^2 S_{ab}^i = tr\left(\Psi^i L^i \left(\Psi^i\right)^T\right) \quad (9)$$

where $L^i = O^i - S^i$, O^i is a diagonal matrix and $O_{aa}^i = \sum_{b=1}^{s} S_{ab}^i$ and $tr(\cdot)$ is the matrix trace operation. Here L^i is a Laplacian matrix [54].

In the particular application of human motion capture, data are prone to containing noises and outliers due to some reasons like markers occlusion and mislabeling. Some motion noise frequently pollutes the data entries within multiple nearby frames due to the reason that once the occluded markers appear, they will last for a short period of time.

In other words, the noise in motion data is strongly structural. It is well known that the ℓ_2-norm is optimal with respect to Gaussian noise distribution, but is sensitive to outliers. The ℓ_1-norm meets the demand for modeling the outliers, but cannot preserve the structure information. As an extension of ℓ_1-norm, the $\ell_{2,1}$-norm that assumes the noise follows the Laplacian distribution is often used to find jointly sparse solutions [43], [55], [56]. Actually, computational studies have showed that the $\ell_{2,p}$-norm provides much more flexible and robustness than the $\ell_{2,1}$-norm [57]. Thus, we enforce the $\ell_{2,p}$-norm penalty on the noise term and rewrite (8) as follows:

$$\min_{\Psi^i,E^i} \left\| Y_t^i - D^i\Psi^i - E^i \right\|_F^2 + \lambda_1 \left\| E^i \right\|_{2,p} + \lambda_2 tr\left(\Psi^i L^i \left(\Psi^i\right)^T\right) + \lambda_3 \left\| \Psi^i \right\|_1, \quad p \in (0,2], \quad i = 1, \ldots, 5. \tag{10}$$

In (10), the sparse coefficients matrix Ψ^i and the noise matrix E^i are jointly estimated in the same framework. In this framework, the noise term, i.e., E^i, models both outliers and the structure noise, while the remained Gaussian noise in $Y_t^i - E^i$ can be perfectly removed by applying the squared Forbenius norm on the first term. Owing to the smooth graph constraint on the third term, the recovered human motion would be much more smooth and stable. The last ℓ_1-norm penalty on Ψ^i brings in the benefit that our method can automatically select a most correlated subset of motion bases from motion dictionaries for clean motion reconstruction. As a result, our method doesn't need to specifically choose the training data set, and it can be very easily applied to real-world applications. Indeed, (10) is a sparse coding problem. Since the noise structure information has been taken into account in (10), we call this model as robust structured sparse coding.

2) Optimization Method: Due to the nonsmooth property of the $\ell_{2,p}$-norm and the nonconvex of (10) with respect to Ψ^i and E^i jointly, it is difficult to solve the problem. Here we apply a coordinate descent scheme for solving this problem. Since each pose is divided into five partlets, it needs to solve five similar objective functions derived from (10). To simplify the notation, we omit the superscript i and try to optimize a common objective function as below

$$\min_{\Psi,E} \|Y_t - D\Psi - E\|_F^2 + \lambda_1 \|E\|_{2,p} + \lambda_2 tr\left(\Psi L \Psi^T\right) + \lambda_3 \|\Psi\|_1. \tag{11}$$

Firstly, we fix E and optimize the above (11) with respect to Ψ. Let $Z = Y_t - E$, (11) is equivalent to

$$\min_{\Psi} \|Z - D\Psi\|_F^2 + \lambda_2 tr\left(\Psi L \Psi^T\right) + \lambda_3 \|\Psi\|_1. \tag{12}$$

Using the matrix vectorization operation and Kronecker product operation, (12) can be rewritten as

$$\begin{aligned} \mathcal{J}(\Psi) &= \|Z - D\Psi\|_F^2 + \lambda_2 tr\left(\Psi L \Psi^T\right) + \lambda_3 \|\Psi\|_1 \\ &\Leftrightarrow tr\left((Z - D\Psi)^T (Z - D\Psi)\right) + \lambda_2 tr\left(\Psi L \Psi^T\right) + \lambda_3 \|\Psi\|_1 \\ &\Leftrightarrow \mathrm{vec}(\Psi)^T \left(I_n \otimes D^T D\right) \mathrm{vec}(\Psi) - 2\mathrm{vec}(\Psi)^T \mathrm{vec}\left(D^T Z\right) \\ &\quad + \lambda_2 \mathrm{vec}(\Psi)^T (L \otimes I_K) \mathrm{vec}(\Psi) + \lambda_3 \|\mathrm{vec}(\Psi)\|_1 \\ &\Leftrightarrow \mathrm{vec}(\Psi)^T A \mathrm{vec}(\Psi) - 2\mathrm{vec}(\Psi)^T \mathbf{b} + \lambda_3 \|\mathrm{vec}(\Psi)\|_1 \end{aligned} \tag{13}$$

where $A = I_n \otimes D^T D + \lambda_2 (L \otimes I_K)$ and $\mathbf{b} = \mathrm{vec}(D^T Z)$. Because both $D^T D$ and L are symmetrical matrices, A is a symmetrical matrix. The eigenvalue decomposition of A can be written as $A = U \Lambda U^T$. Therefore, the first two terms in (13) can be rewritten as

$$\begin{aligned} &\mathrm{vec}(\Psi)^T A \mathrm{vec}(\Psi) - 2\mathrm{vec}(\Psi)^T \mathbf{b} \\ &\quad = \mathrm{vec}(\Psi)^T \left(\Lambda^{\frac{1}{2}} U^T\right)^T \left(\Lambda^{\frac{1}{2}} U^T\right) \mathrm{vec}(\Psi) \\ &\qquad - 2\mathrm{vec}(\Psi)^T \left(\Lambda^{\frac{1}{2}} U^T\right)^T \left(\left(\Lambda^{\frac{1}{2}} U^T\right)^T\right)^{-1} \mathbf{b} \\ &\quad \Leftrightarrow \left\| \left(\Lambda^{\frac{1}{2}} U^T\right) \mathrm{vec}(\Psi) - \left(\left(\Lambda^{\frac{1}{2}} U^T\right)^T\right)^{-1} \mathbf{b} \right\|_2^2. \end{aligned} \tag{14}$$

According to (13) and (14), if we denote $\mathbf{A}^* = \Lambda^{\frac{1}{2}} U^T$ and $\mathbf{b}^* = ((\Lambda^{\frac{1}{2}} U^T)^T)^{-1} \mathbf{b}$, (12) is equivalent to

$$\min_{\mathrm{vec}(\Psi)} \left\| \mathbf{A}^* \mathrm{vec}(\Psi) - \mathbf{b}^* \right\|_2^2 + \lambda_3 \|\mathrm{vec}(\Psi)\|_1. \tag{15}$$

Indeed, (15) is a standard ℓ_1 least square problem, that is the lasso problem. The least angle regression algorithm and its variants can be applied to find the optimal solution of (15). It has to point out that since we vectorize Ψ, the size of the matrix $\mathbf{A}^*$ may be very large, which is time consuming to solve (15). In this case, we can directly solve (12) using the first order method such as proximal gradient method [58].

Secondly, when the optimal value of Ψ has been found, we optimize (11) with respect to E and obtain

$$\min_{E} \|Y_t - D\Psi - E\|_F^2 + \lambda_1 \|E\|_{2,p}. \tag{16}$$

Let $Q = Y_t - D\Psi$ and define a diagonal matrix Θ with its diagonal elements $\Theta_{ii} = 1/(2/p)\|E_{i\cdot}\|_2^{2-p}$, then (16) becomes

$$\min_{E} \|Q - E\|_F^2 + \lambda_1 tr\left(E^T \Theta E\right). \tag{17}$$

By setting the derivative with respect to E to 0, we get

$$E = (I + \lambda_1 \Theta)^{-1} Q. \tag{18}$$

Since Θ is a diagonal matrix, the inverse matrix of $I + \lambda_1 \Theta$, i.e., $(I + \lambda_1 \Theta)^{-1}$, can be effectively calculated. Thus, it is easy to calculate E based on (17).

Till now, we obtain the two updating rules for both Ψ and E. These updating rules should be recursively applied until the convergence is achieved and a local optimum solution of Ψ and E is obtained. Recall that each human pose is divided into five partlets, here we solve five similar objective functions derived from (10) in parallel.

3) Motion Reconstruction: After solving the above (11), we get the sparse representation coefficient matrix Ψ^i, $i = 1, \ldots, 5$. Now, we can reconstruct the filtered clean group motion matrix via $\tilde{Y}_t^i = D^i \Psi^i$. Recall that $Y_t^i = [\mathbf{g}_1^i, \ldots, \mathbf{g}_s^i], s = n - m + 1$ and $\mathbf{g}_j^i = \mathrm{vec}([X_{\cdot j}^i, X_{\cdot j+1}^i, \ldots, X_{\cdot j+m-1}^i]) \in \mathcal{R}^{(d_i \times m) \times 1}$. So, we decompose the filtered partlets groups $\tilde{\mathbf{g}}_j^i$ in $\tilde{Y}_t^i$ and calculate the mean value for each partlet, e.g., $\tilde{X}_{\cdot j}^i = (1/n_j) \sum_{t=1}^{n_j} (\tilde{X}_{\cdot j}^i)_t$ wherein $(\tilde{X}_{\cdot j}^i)_t$ is the tth copy of $\tilde{X}_{\cdot j}^i$ and n_j is the total number of the copy of the partlet $\tilde{X}_{\cdot j}^i$ in $\tilde{Y}_t^i$. Because the ith partlet

Algorithm 1 Human Motion Denoising Algorithm

Input: five motion dictionary matrices:$D^i, i = 1, \ldots, 5$ learned by the robust dictionary learning algorithm; the noisy global motion sequence:X_{global};the length of lagged window:m; regularized parameters: λ_1, λ_2 and λ_3; the value of p.

Output: the filtered global motion sequence: $\tilde{X}_{\text{global}}$.

1: **Coordinate Normalization**:
normalize and translate the global noisy motion sequence X_{global} into the local noisy motion sequence X_{local};

2: **Partlet Generation:**
generate partlets $X^i = [X^i_{\cdot 1}, \ldots, X^i_{\cdot n}] \in \mathcal{R}^{d_i \times n}, i = 1, \ldots, 5$ based on X_{local}

3: **Partlet Grouping:**
generate partlets-groups $Y^i_t = [\mathbf{g}^i_1, \ldots, \mathbf{g}^i_s], s = n - m + 1$ where $\mathbf{g}^i_j = \text{vec}([X^i_{\cdot j}, X^i_{\cdot j+1}, \ldots, X^i_{\cdot j+m-1}])$ according to $X^i, i = 1, \ldots, 5$.

4: **Motion Denoising:**
Parfor $i = 1{:}5$
Step1:
Initialize E^i and compute L^i;
Repeat
Fix E^i and update Ψ^i according to (15);
Fix Ψ^i and update E^i according to (17);
Until Convergence;
Compute the filtered human motion $\tilde{Y}^i_t = D^i \Psi^i$;
End

5: **Decompose partlet Groups:**
decompose $\tilde{Y}^i_t, i = 1, \ldots, 5$ to obtain multiple partlets.

6: **Calculate Filtered partlets:**
calculate the mean value for each partlet, e.g., $\tilde{X}^i_{\cdot j} = \frac{1}{n_j} \sum_{t=1}^{n_j} (\tilde{X}^i_{\cdot j})_t$ wherein $(\tilde{X}^i_{\cdot j})_t$ is the t-th copy of $\tilde{X}^i_{\cdot j}$ and n_j is the total number of copy of the partlet $\tilde{X}^i_{\cdot j}$ in $\tilde{Y}^i_t$.

7: **Form Local Motion Matrix:**
form the filtered submatrix $\tilde{X}^i$ based on $\tilde{X}^i_{\cdot j}$, and then obtain the filtered local motion matrix $\tilde{X}_{\text{local}}$.

8: **Coordinate Transformation**:
convert the local filtered motion sequence $\tilde{X}_{\text{local}}$ into the global filtered motion sequence $\tilde{X}_{\text{global}}$.

is $X^i = [X^i_{\cdot 1}, \ldots, X^i_{\cdot n}] \in \mathcal{R}^{d_i \times n}, i = 1, \ldots, 5$, we can recover the filtered submatrix $\tilde{X}^i$ based on the recovered partlet $\tilde{X}^i_{\cdot j}$. It is also easy to form the local motion matrix $\tilde{X}_{\text{local}}$. Finally, we translate the local poses to be the global poses $\tilde{X}_{\text{global}}$ according to the recorded transformation matrix M to achieve the goal of human motion de-noising. Here, we summarize our proposed human motion denoising approach in Algorithm 1.

IV. Experiments

The performance of our proposed approach was evaluated on both the simulated and real noisy motion data. To quantitatively assess the performance of our algorithm, we first compare it with other four widely used human motion denoising algorithms with the simulated data, which include a variety of motion noises. Then, we apply these denoising algorithms to deal with the real noisy human motion data captured by a commercial optical motion capture system (i.e., MotionAnalysis Raptor-E Digital RealTime System) and a Microsoft Kinect. Since, there are several model parameters such as λ_1, λ_2, and λ_3 in (10) in our algorithm, we finally conduct a series of experiments to study the parameter sensitivity of our approach.

A. Testing on Simulated Data

Since the algorithm's performance may be affected by multiple factors including the complexity of action, the noise type and noise level, we select more than 80 motion sequences including two simple actions (i.e., walk and jump) and two complex actions (i.e., dance and boxing) from CMU human motion database [20]. We used the asf/amc files, which contain 32 markers, in our experiments. Because most data in the CMU database are clean, we randomly select 2 sequences of each action to synthesize three kinds of noise: 1) Gaussian noise with signal-to-noise ratio (SNR) ranges from {30, 25, 20, 15, 10, 5} dB; 2) outlier with ratio from 5% to 30% with an interval of 5%; and 3) mixed noise that consists of some Gaussian noise and outliers. The remainder motion sequences are used for training. We compare our method with the following algorithms: 1) Gaussian filter; 2) Wavelet filter [16]; 3) Kalman filter [31]; and 4) the example-based method [17]. The first three methods are widely used in commercial motion capture systems, while the last one is a well-known data-driven-based human motion denoising method.

We apply a Gaussian filter, Wavelet filter, and Kalman filter to denoise each feature dimension of noisy motion data independently. For our method and the example-based method, we use the clean motion sequences to train algorithm models and test them with the noisy motion data. For fair comparison, we tune all model parameters for each algorithm and report their best results. Take the example-based method for example, we tune the size of lagged window from {5, 10, 15, 20, 25, 30, 35, 40} and the number of reserved bases K from {20, 40, 60, 80, 100, 120} and choose the best setting via cross-validation. For our method, we let the motion bases number, i.e., $K_i, i = 1, \ldots, 5$, to be the same value for simplicity and denote them as K to be consist with [17]. We empirically set the two parameters λ and β used in the robust dictionary learning algorithm to $\lambda = 10^{-3}$ and $\beta = 10^{-1}$. The regularized parameters in the robust structured sparse coding model like λ_1, λ_2, and λ_3 are tuned from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^2, 10^3\}$. The sparseness parameter p is in $(0, 2]$, so we tune it from {0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2}.

To quantify the de-noised results, the root mean squared error (RMSE) measurement [19], [37]

$$\text{RMSE}\left(X, \tilde{X}\right) = \sqrt{\frac{\sum_{i=1}^{n} \left(\sum_{j=1}^{l} \left\| P_{ji} - \widetilde{P}_{ji} \right\|^2 \right)}{n \times l}} \tag{19}$$

where P_{ji} is the jth marker in the ith pose in clean motion data and $\widetilde{P}_{ji}$ is the corresponding filtered marker, is used in this paper.

Due to the space limitation, we show the results of only one sequence for each action. The selected sequences are 08_11 (walk), 02_04 (jump), 05_15 (dance), and 17_10 (boxing). From Tables I–III, we can see that our

TABLE I
SUMMARY OF PERFORMANCE FOR THE COMPARED ALGORITHMS ON CMU HUMAN MOTION DATA WITH VARIOUS GAUSSIAN NOISE. THE RMSE (CM/MARKER) AND STANDARD DEVIATIONS ARE REPORTED. THE HIGHEST PERFORMANCE IS HIGHLIGHTED IN EACH CASE

Action	Algorithm	SNR=30dB	SNR=25dB	SNR=20dB	SNR=15dB	SNR=10dB	SNR=5dB
Walk	Our	$\mathbf{0.88 \pm 0.10}$	$\mathbf{1.46 \pm 0.15}$	$\mathbf{1.61 \pm 0.20}$	$\mathbf{1.78 \pm 0.24}$	$\mathbf{2.26 \pm 0.36}$	$\mathbf{3.17 \pm 0.42}$
	Example-based	1.39 ± 0.15	2.48 ± 0.64	2.60 ± 0.55	3.67 ± 0.68	4.37 ± 0.80	5.07 ± 1.05
	Gaussain	1.02 ± 0.31	2.85 ± 0.83	2.02 ± 0.34	3.72 ± 0.84	4.96 ± 0.67	7.69 ± 1.03
	Kalman	2.18 ± 0.67	2.32 ± 0.67	2.77 ± 0.68	3.54 ± 0.73	5.14 ± 0.95	8.74 ± 2.15
	Wavelet	1.23 ± 0.32	1.85 ± 0.41	2.77 ± 0.68	3.97 ± 0.84	5.92 ± 1.18	9.63 ± 1.64
Jump	Our	$\mathbf{0.83 \pm 0.18}$	$\mathbf{1.61 \pm 0.79}$	$\mathbf{1.73 \pm 0.73}$	$\mathbf{1.99 \pm 0.67}$	$\mathbf{2.56 \pm 0.59}$	$\mathbf{3.59 \pm 0.89}$
	Example-based	1.34 ± 0.13	2.05 ± 0.87	2.34 ± 0.55	3.46 ± 1.02	3.92 ± 1.31	4.70 ± 1.45
	Gaussain	1.27 ± 0.88	3.36 ± 2.71	2.18 ± 0.67	4.22 ± 2.52	5.22 ± 1.02	7.90 ± 1.92
	Kalman	2.81 ± 2.75	2.66 ± 1.93	3.05 ± 1.81	3.84 ± 1.74	5.71 ± 2.32	8.55 ± 1.76
	Wavelet	1.21 ± 0.44	1.80 ± 0.65	2.70 ± 1.01	3.97 ± 1.57	5.99 ± 2.17	8.56 ± 3.08
Dance	Our	$\mathbf{0.86 \pm 0.18}$	$\mathbf{2.37 \pm 0.73}$	$\mathbf{2.55 \pm 0.71}$	$\mathbf{2.87 \pm 0.70}$	$\mathbf{3.73 \pm 0.83}$	$\mathbf{5.23 \pm 1.05}$
	Example-based	1.58 ± 0.21	2.51 ± 0.78	2.81 ± 0.69	4.44 ± 1.11	6.00 ± 1.91	5.95 ± 1.62
	Gaussain	2.13 ± 0.68	5.87 ± 2.01	2.96 ± 0.56	6.72 ± 1.88	6.65 ± 1.06	10.53 ± 1.40
	Kalman	4.39 ± 1.57	4.49 ± 1.54	4.21 ± 1.36	5.73 ± 1.47	7.24 ± 1.35	11.11 ± 2.15
	Wavelet	1.87 ± 0.44	2.66 ± 0.63	4.03 ± 0.96	6.04 ± 1.51	8.60 ± 2.11	12.73 ± 2.71
Boxing	Our	$\mathbf{0.85 \pm 0.20}$	$\mathbf{1.38 \pm 0.35}$	$\mathbf{1.49 \pm 0.25}$	$\mathbf{2.10 \pm 0.35}$	$\mathbf{3.22 \pm 0.71}$	$\mathbf{4.33 \pm 0.79}$
	Example-based	1.41 ± 0.12	1.54 ± 0.51	2.12 ± 0.32	3.11 ± 0.64	4.83 ± 0.94	4.20 ± 0.88
	Gaussain	1.78 ± 0.86	2.64 ± 0.78	3.89 ± 1.84	4.76 ± 1.71	5.84 ± 1.00	8.61 ± 1.46
	Kalman	2.84 ± 1.35	3.01 ± 1.34	3.46 ± 1.29	4.22 ± 1.20	5.81 ± 1.09	9.10 ± 1.15
	Wavelet	1.78 ± 0.46	2.48 ± 0.67	3.46 ± 0.97	4.86 ± 1.37	6.75 ± 1.80	9.24 ± 2.25

TABLE II
SUMMARY OF PERFORMANCE FOR THE COMPARED ALGORITHMS ON CMU HUMAN MOTION DATA WITH VARIOUS OUTLIER NOISE. THE RMSE (CM/MARKER) AND STANDARD DEVIATIONS ARE REPORTED. THE HIGHEST PERFORMANCE IS HIGHLIGHTED IN EACH CASE

Action	Algorithm	Ratio=5%	Ratio=10%	Ratio=15%	Ratio=20%	Ratio=25%	Ratio=30%
Walk	Our	$\mathbf{1.43 \pm 0.19}$	$\mathbf{1.46 \pm 0.22}$	$\mathbf{1.68 \pm 0.26}$	$\mathbf{1.84 \pm 0.26}$	$\mathbf{1.88 \pm 0.26}$	$\mathbf{1.98 \pm 0.36}$
	Example-based	1.50 ± 0.35	1.63 ± 0.35	1.81 ± 0.52	1.99 ± 0.56	2.17 ± 0.77	2.66 ± 0.88
	Gaussain	1.46 ± 0.39	2.18 ± 0.52	2.77 ± 0.46	3.68 ± 0.51	4.64 ± 0.68	3.36 ± 0.60
	Kalman	2.30 ± 0.69	2.69 ± 0.80	3.01 ± 0.67	3.76 ± 0.70	4.56 ± 0.86	5.27 ± 0.74
	Wavelet	1.82 ± 0.64	2.80 ± 0.78	3.46 ± 0.81	4.21 ± 1.00	5.14 ± 1.16	5.80 ± 1.25
Jump	Our	1.33 ± 0.39	1.47 ± 0.37	$\mathbf{1.50 \pm 0.36}$	$\mathbf{1.70 \pm 0.49}$	$\mathbf{1.74 \pm 0.33}$	$\mathbf{1.78 \pm 0.27}$
	Example-based	$\mathbf{1.15 \pm 0.63}$	$\mathbf{1.43 \pm 0.49}$	1.50 ± 0.48	1.91 ± 0.72	1.96 ± 1.76	2.31 ± 0.83
	Gaussain	1.71 ± 0.79	2.42 ± 0.76	3.14 ± 0.60	3.91 ± 0.75	4.76 ± 0.61	5.87 ± 0.71
	Kalman	2.78 ± 1.88	3.15 ± 1.67	3.73 ± 1.43	4.38 ± 1.44	5.13 ± 1.21	6.04 ± 1.04
	Wavelet	1.75 ± 0.66	2.98 ± 1.12	6.68 ± 1.35	4.39 ± 1.46	5.23 ± 1.68	6.20 ± 1.66
Dance	Our	$\mathbf{1.47 \pm 0.44}$	$\mathbf{1.67 \pm 0.46}$	$\mathbf{1.72 \pm 0.44}$	$\mathbf{1.93 \pm 0.47}$	$\mathbf{2.10 \pm 0.47}$	$\mathbf{2.41 \pm 0.44}$
	Example-based	1.82 ± 0.47	1.90 ± 0.48	2.08 ± 0.67	2.20 ± 0.59	2.31 ± 0.71	2.42 ± 0.84
	Gaussain	2.44 ± 0.66	3.13 ± 0.51	4.10 ± 0.61	4.98 ± 0.84	5.96 ± 0.76	7.25 ± 0.87
	Kalman	4.31 ± 1.44	4.68 ± 1.35	5.26 ± 1.21	5.90 ± 1.14	6.61 ± 1.19	7.57 ± 1.09
	Wavelet	2.37 ± 0.78	3.99 ± 0.88	5.10 ± 1.15	6.01 ± 1.42	6.94 ± 1.46	7.97 ± 1.52
Boxing	Our	$\mathbf{1.00 \pm 0.27}$	$\mathbf{1.26 \pm 0.29}$	$\mathbf{1.07 \pm 0.35}$	$\mathbf{1.55 \pm 0.37}$	$\mathbf{1.85 \pm 0.45}$	$\mathbf{2.26 \pm 0.55}$
	Example-based	1.37 ± 0.30	1.59 ± 0.29	1.99 ± 0.38	2.16 ± 0.56	4.02 ± 0.55	4.55 ± 0.64
	Gaussain	2.25 ± 0.79	3.03 ± 0.72	3.94 ± 0.72	4.70 ± 0.68	5.65 ± 0.66	6.77 ± 0.84
	Kalman	3.14 ± 1.26	3.73 ± 1.18	4.47 ± 1.10	5.14 ± 1.01	6.00 ± 0.91	6.84 ± 0.86
	Wavelet	2.39 ± 0.67	3.84 ± 0.89	4.66 ± 1.03	5.50 ± 1.06	6.38 ± 1.09	7.25 ± 1.10

method consistently outperforms its competitors. More importantly, the standard deviations of our method are mostly smaller than the others, which means that the outputs of our method are much more stable than that of the others.

B. Testing on Real Data

In the real data experiments, we first capture a variety of actions like walk, jump, boxing, hugging, and picking-up performing by two subjects using a MotionAnalysis Raptor-E Digital RealTime System, and each action repeats five times. These motion data contain 42 markers in each pose. As mentioned before, the acquired raw motion data often contain a certain percentage of missing values. So before the experiments we manually label all of the unnamed markers using the post-processing tool provided by the motion capture system and just apply the spline interpolation method to fill the remainder missing values. The complete motion data are then

TABLE III
SUMMARY OF PERFORMANCE FOR THE COMPARED ALGORITHMS ON CMU HUMAN MOTION DATA WITH VARIOUS MIXED NOISE (THE SNR OF GAUSSIAN NOISE AND THE RATIO OF OUTLIER ARE SHOWN IN THE HEADLINE OF TABLE). THE RMSE (CM/MARKER) AND STANDARD DEVIATIONS ARE REPORTED. THE HIGHEST PERFORMANCE IS HIGHLIGHTED IN EACH CASE

Action	Algorithm	$20dB + 10\%$	$20dB + 20\%$	$10dB + 10\%$	$10dB + 20\%$
Walk	Our	$\mathbf{1.83 \pm 0.26}$	$\mathbf{2.54 \pm 0.42}$	$\mathbf{3.00 \pm 0.44}$	$\mathbf{3.01 \pm 0.39}$
	Example-based	2.79 ± 0.63	3.40 ± 0.77	5.69 ± 1.27	5.75 ± 0.93
	Gaussain	2.96 ± 0.50	4.48 ± 0.99	6.13 ± 0.87	6.38 ± 0.85
	Kalman	3.18 ± 0.76	4.31 ± 0.81	6.50 ± 0.87	6.79 ± 1.04
	Wavelet	3.73 ± 0.77	4.87 ± 0.97	7.13 ± 1.18	7.41 ± 1.24
Jump	Our	$\mathbf{1.91 \pm 0.45}$	$\mathbf{2.07 \pm 0.76}$	$\mathbf{2.72 \pm 0.44}$	$\mathbf{2.84 \pm 0.67}$
	Example-based	2.48 ± 0.74	2.99 ± 1.18	4.20 ± 1.41	5.20 ± 1.33
	Gaussain	3.16 ± 0.58	5.29 ± 2.00	5.85 ± 2.15	6.68 ± 1.84
	Kalman	3.67 ± 1.44	4.86 ± 1.29	5.95 ± 1.51	6.75 ± 1.10
	Wavelet	3.74 ± 1.04	5.07 ± 1.58	6.23 ± 2.25	7.21 ± 2.32
Dance	Our	$\mathbf{2.42 \pm 0.53}$	$\mathbf{2.92 \pm 0.65}$	$\mathbf{3.67 \pm 0.59}$	$\mathbf{3.91 \pm 0.78}$
	Example-based	2.99 ± 0.82	3.70 ± 1.08	5.35 ± 1.30	6.92 ± 1.46
	Gaussain	3.96 ± 0.63	7.35 ± 1.67	8.12 ± 1.74	9.01 ± 1.60
	Kalman	5.20 ± 1.23	6.37 ± 1.10	8.03 ± 2.82	8.63 ± 1.15
	Wavelet	5.37 ± 1.17	6.85 ± 1.31	9.23 ± 2.28	10.31 ± 1.86
Boxing	Our	$\mathbf{2.04 \pm 0.41}$	$\mathbf{2.17 \pm 0.41}$	$\mathbf{3.42 \pm 0.52}$	$\mathbf{3.58 \pm 0.53}$
	Example-based	2.19 ± 0.43	2.51 ± 0.66	4.94 ± 0.91	5.80 ± 0.94
	Gaussain	3.79 ± 0.72	6.12 ± 1.51	6.56 ± 1.57	7.64 ± 1.42
	Kalman	4.24 ± 1.15	5.58 ± 1.02	6.51 ± 1.12	7.61 ± 0.99
	Wavelet	4.74 ± 1.10	6.19 ± 1.23	7.31 ± 1.95	8.38 ± 1.81

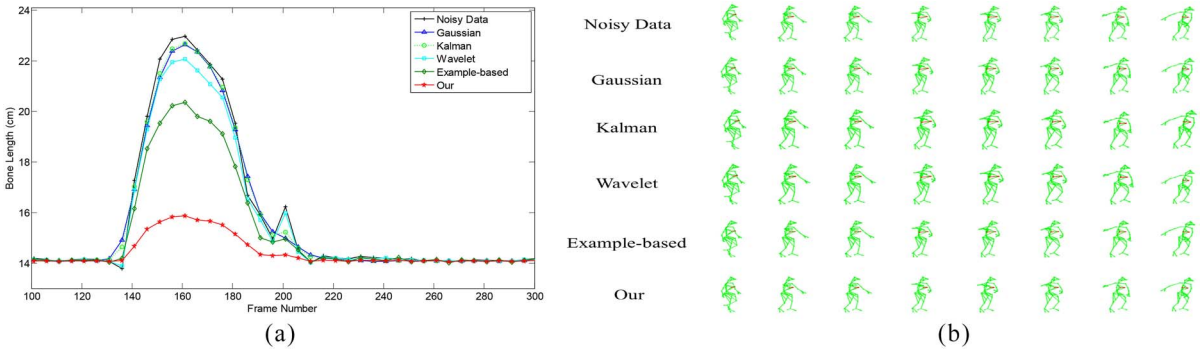


Fig. 6. Comparison between different algorithms on a hugging motion sequence. (a) Length variant curves for a chest bone in the original noisy motion sequence as well as the results of different algorithms. (b) Key poses on frame numbers 111, 139, 153, 167, 181, 195, 209, and 223 in a hugging motion sequence. From the top row to the bottom row, the key poses of the original noisy motion sequence and the results of different algorithms are presented one-by-one.

adopted in our experiments. For each action, we randomly select one sequence as the testing data, while the rest nine sequences are used as the training data for the example-based method [17]. In contrast, for our method we simply remove the testing sequence and use all the other motion sequences as the training data.

Because we do not have any ground truth of the testing data, we compare the bone-length of the outputs of different algorithms. In Figs. 6 and 7, one bone of human body is selected and marked with red color. The output of our method is more stable than that of its competitors. We note that our method outperforms the example-based method. We believe it is because the proposed robust dictionary learning algorithm as well as the robust structured sparse coding algorithm make our approach become more robust against with noises and outliers.

To demonstrate the effectiveness of our method, we apply the proposed approach to deal with the outputs from a Microsoft Kinect [22]. The pose acquired from a Microsoft Kinect comprises only 20 joints, which is less than that comes from an optical motion capture system. Similarly, we capture five motion sequences for each action and then adopt the same experimental setting as that is used in dealing with the real optical motion capture data. In Fig. 8, we show the original imperfect skeletons and the denoising results of our method in two motion sequences. The human poses of our algorithm are much more stable and correct than the raw data. It demonstrates that our method can be used to refine the real imperfect motion data.

C. Parameter Sensitivity and Convergence

In addition, we conduct experiments to study the parameter sensitivity of our algorithm using the simulated data. Fig. 9 shows that p should be carefully set under different noise condition. In other words, it is important to take the noise structure information into account. In Fig. 10, we study the

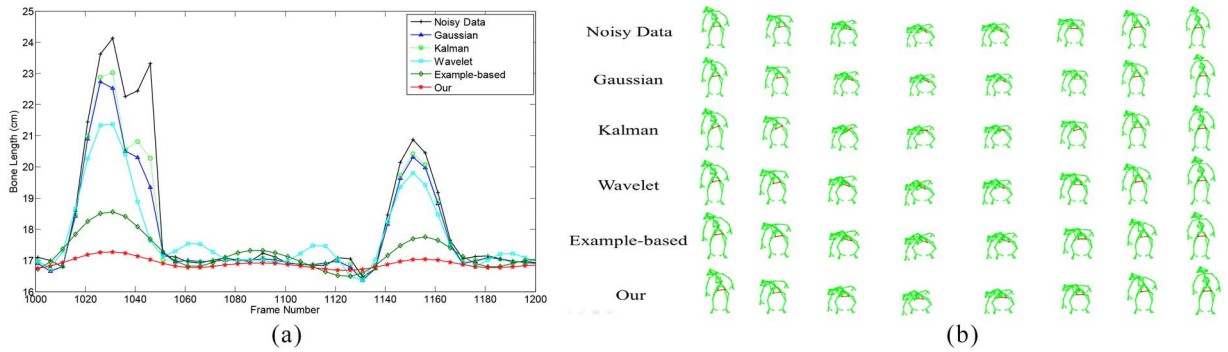


Fig. 7. Comparison between different algorithms on a picking-up motion sequence. (a) Length variant curves for a bone connected the left and right femur joints in the original noisy motion sequence as well as the results of different algorithms. (b) Key poses on frame numbers 1111, 1123, 1135, 1147, 1159, 1171, 1183, and 1195 in a picking-up motion sequence. From the top row to the bottom row, the key poses of the original noisy motion sequence and the results of different algorithms are presented one-by-one.

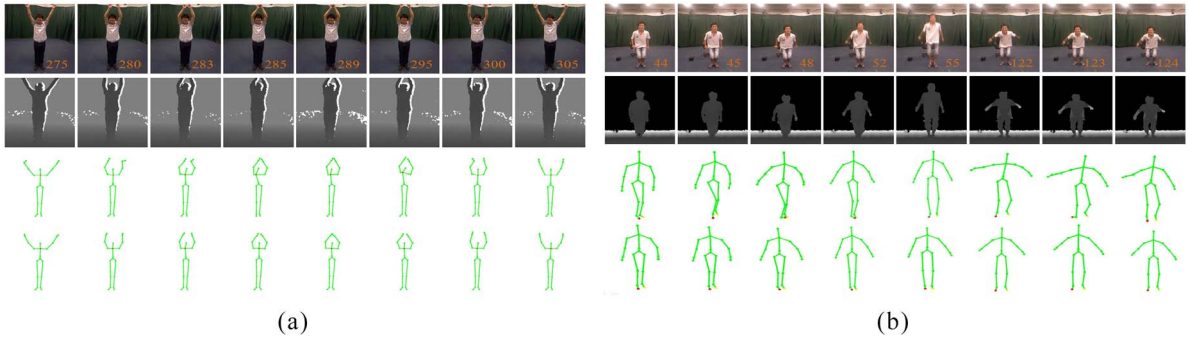


Fig. 8. Denoising motion capture data with a Microsoft Kinect using our proposed method. From the top row to bottom row, the original captured color images, depth images, skeletons using Kinect, as well as the denoising results of our method are presented one-by-one. (a) Raise hands. (b) Jump.

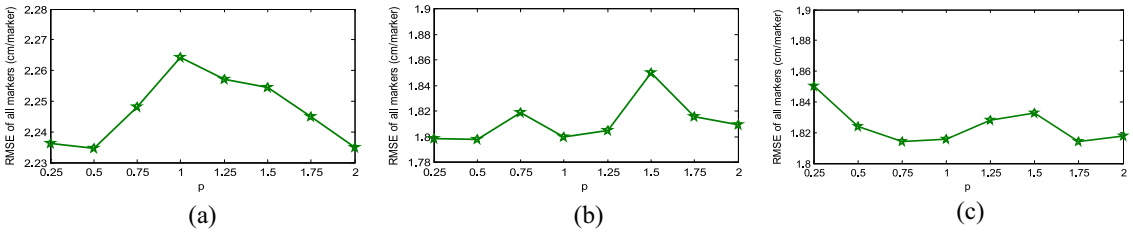


Fig. 9. Performance variance of our method with respect to the p-value on three walk motion sequences, which contain (a) Gaussian noise (10 dB), (b) outlier (20%), and (c) mixed noise (20 dB + 10%), respectively.

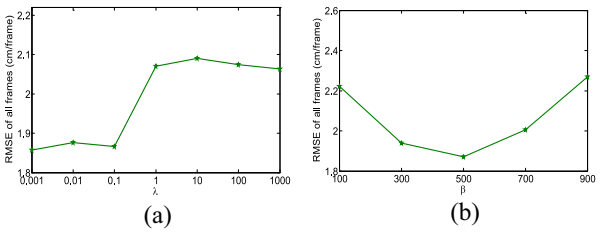


Fig. 10. Performance variance of our method with respect to (a) λ and (b) β.

performance variance of our method with respect to the parameters of robust dictionary learning model, i.e., λ and β. In this testing, we fixed $\lambda_1 = 10^2$, $\lambda_2 = 10^{-3}$, and $\lambda_3 = 10^{-2}$. As we can see, the smaller the λ value, the better performance of our model. When $\lambda \to \infty$, our dictionary learning model [i.e., (5)] is reduced to the traditional dictionary learning algorithm like K-SVD [59]. In other words, we should take the outliers into account in learning the multiple motion dictionaries. Meanwhile, Fig. 10(b) shows that we should carefully tune β, so W is not too dense or too sparse, which will decrease the algorithm's performance. In Fig. 11, we find when λ_1, λ_2, and λ_3 are smaller than 1, the RMSE value is acceptable. Relatively speaking, our method is much more sensitive with respect to λ_2 and λ_3 than λ_1. From Fig. 12, we find that the bigger dictionary size and window size, the better performance in a certain range. But it needs more time to solve the objective function. And from Table IV, we can see that the partlet representation not only reduces the entire data processing time, but also improves the performance of our method. Lastly, Fig. 13 shows the convergence curves of our optimization algorithms for solving the robust dictionary learning model and robust structured sparse coding model. As shown in Fig. 13, it converges within 30 and 100 iterations, respectively, in solving the proposed two models.

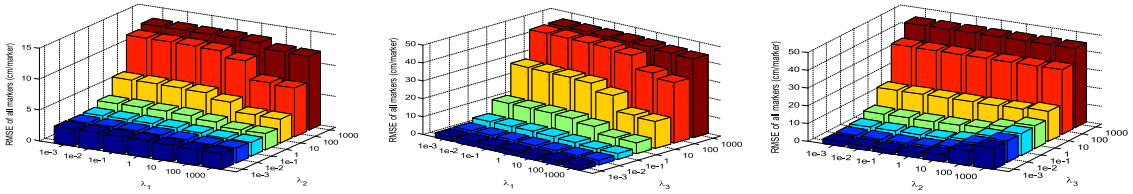


Fig. 11. Performance variance of our method with respect to the parameters λ_1, λ_2, and λ_3 on a noisy walk motion sequence. (a) λ_1 and λ_2 with fixed $\lambda_3 = 0.01$. (b) λ_1 and λ_3 with fixed $\lambda_2 = 10^{-3}$. (c) λ_2 and λ_3 with fixed $\lambda_1 = 10^2$.

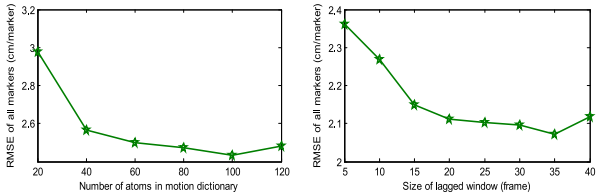


Fig. 12. Performance variance of our method with respect to the (a) dictionary size and (b) lagged window size.

TABLE IV
PERFORMANCE COMPARISON BETWEEN OUR METHOD WITH AND WITHOUT USING THE PARTLET MODEL REPRESENTATION UNDER DIFFERENT NOISE CONDITIONS USING THE WALK MOTION SEQUENCE (08_11). HERE GAUSSIAN NOISE: *SNR* = 10 dB, OUTLIER NOISE: *Ratio* = 20%, AND MIXED NOISE: 20 dB + 10%

Noise	Representation	RMSE(cm/marker)	Time(s)
Gaussian	Pose	3.00 ± 0.52	28.95
	partlets	2.26 ± 0.36	13.74
Outlier	Pose	2.85 ± 0.45	27.46
	partlets	1.88 ± 0.28	14.01
Mixed	Pose	2.93 ± 0.45	28.76
	partlets	1.90 ± 0.26	13.75

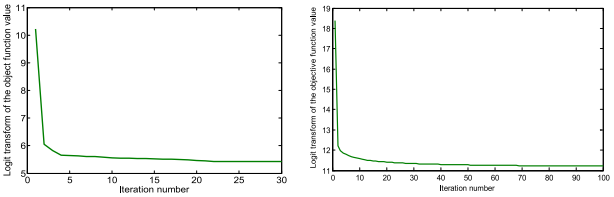


Fig. 13. Convergence curves of our proposed optimization methods for solving the (a) robust dictionary learning model and (b) robust structured sparse coding model.

So, the proposed optimization algorithms converge very fast in our application.

V. CONCLUSION

Human motion denoising is an indispensable step for motion data processing. We have proposed a new data-driven-based robust human motion denoising approach for removing both the noise and outliers. Experiments on both the simulated and real human motion data show that our method consistently yields better performance than other methods. The outputs of our method are much more stable than the others. And, it is very easy to setup the training dataset for our method.

REFERENCES

[1] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Comput. Vis. Image Understand.*, vol. 81, no. 3, pp. 231–268, 2001.
[2] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Comput. Vis. Image Understand.*, vol. 104, no. 2, pp. 90–126, 2006.
[3] M. Tejera, D. Casas, and A. Hilton, "Animation control of surface motion capture," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1532–1545, Jun. 2013.
[4] F. Zhu and L. Shao, "Weakly-supervised cross-domain dictionary learning for visual recognition," *Int. J. Comput. Vis.*, vol. 109, no. 1–2, pp. 42–59, 2014.
[5] S. Jones and L. Shao, "Unsupervised spectral dual assignment clustering of human action in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 604–611.
[6] L. Shao, S. Jones, and X. Li, "Efficient search and localization of human actions in video databases," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 504–512, Aug. 2014.
[7] X. Zhen, L. Shao, and X. Li, "Action recognition by spatio-temporal oriented energies," *Inf. Sci.*, vol. 281, pp. 295–309, Oct. 2014.
[8] M. Gleicher, "Animation from observation: Motion capture and motion editing," *ACM SIGGRAPH Comput. Graph.*, vol. 33, no. 4, pp. 51–54, Nov. 1999.
[9] W. Geng and G. Yu, "Reuse of motion capture data in animation: A review," in *Proc. Int. Conf. Comput. Sci. Appl. III*, New York, NY, USA: Springer, Jun. 2003, pp. 620–629.
[10] D. Vlasic *et al.*, "Practical motion capture in everyday surroundings," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 1–9, Jul. 2007.
[11] J. Tautges *et al.*, "Motion reconstruction using sparse accelerometer data," *ACM Trans. Graph.*, vol. 30, no. 3, pp. 18:1–18:12, May 2011.
[12] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, Jun. 2013.
[13] H. P. Shum, E. S. Ho, Y. Jiang, and S. Takagi, "Real-time posture reconstruction for Microsoft Kinect," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1357–1369, Aug. 2013.
[14] J. Lee and S. Shi, "General construction of time-domain filters for orientation data," *IEEE Trans. Vis. Comput. Graph.*, vol. 8, no. 2, pp. 119–128, Jun. 2002.
[15] T. Tangkuampien and D. Suter, "Human motion de-noising via greedy kernel principal component analysis filtering," in *Proc. 18th Int. Conf. Pattern Recogn. (ICPR)*, vol. 3. Hong Kong, 2006, pp. 457–460.
[16] C.-C. Hsieh and P.-L. Kuo, "An impulsive noise reduction agent for rigid body motion data using B-spline wavelets," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1733–1741, Apr. 2008.
[17] H. Lou and J. Chai, "Example-based human motion denoising," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 5, pp. 870–879, Feb. 2010.
[18] R. Lai, P. Yuen, and K. Lee, "Motion capture data completion and denoising by singular value thresholding," in *Proc. Eurograph.*, 2011, pp. 45–48.
[19] Y. Feng *et al.*, "Exploiting temporal stability and low-rank structure for motion capture data refinement," *Inf. Sci.*, vol. 277, no. 1, pp. 777–793, 2014.
[20] *Carnegie Mellon University Graphics Lab Motion Capture Database*. [Online]. Available: http://mocap.cs.cmu.edu

[21] X. Wei, P. Zhang, and J. Chai, "Accurate realtime full-body motion capture using a single depth camera," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 188:1–188:12, 2012.

[22] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[23] Y. Fangt, C. Hsieh, M. Kim, J. Chang, and T. Woo, "Real time motion fairing with unit quaternions," *Comput.-Aided Design*, vol. 30, no. 3, pp. 192–198, 1998.

[24] A. M. Wink and J. B. Roerdink, "Denoising functional MR images: A comparison of wavelet denoising and Gaussian smoothing," *IEEE Trans. Med. Imag.*, vol. 23, no. 3, pp. 374–387, Mar. 2004.

[25] L. Shao, R. Yan, X. Li, and Y. Liu, "From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1001–1013, Jul. 2014.

[26] I. Akhter, T. Simon, S. Khan, I. Matthews, and Y. Sheikh, "Bilinear spatiotemporal basis models," *ACM Trans. Graph.*, vol. 31, no. 2, pp. 17:1–17:12, 2012.

[27] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[28] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang, "Learning with ℓ_1-graph for image analysis," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 858–866, Apr. 2010.

[29] D. Tao, L. Jin, Z. Yang, and X. Li, "Rank preserving sparse learning for Kinect based scene classification," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1406–1417, Oct. 2013.

[30] K. Yamane and Y. Nakamura, "Dynamics filter-concept and implementation of online motion generator for human figures," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 1. San Francisco, CA, USA, 2000, pp. 688–694.

[31] H. J. Shin, J. Lee, S. Y. Shin, and M. Gleicher, "Computer puppetry: An importance-based approach," *ACM Trans. Graph.*, vol. 20, no. 2, pp. 67–94, 2001.

[32] L. Li, J. McCann, N. S. Pollard, and C. Faloutsos, "BoLeRO: A principled technique for including bone length constraints in motion capture occlusion filling," in *Proc. Symp. Comput. Anim. (SCA)*, 2010, pp. 125–135.

[33] L. Liu, L. Shao, X. Zhen, and X. Li, "Learning discriminative key poses for action recognition," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1860–1870, Dec. 2013.

[34] L. Shao, X. Zhen, D. Tao, and X. Li, "Spatio-temporal Laplacian pyramid coding for action recognition," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 817–827, Jun. 2014.

[35] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3-D Digit. Imag. Model.*, 2001, pp. 145–152.

[36] B. Jan, K. Björn, Z. Arno, and W. Andreas, "Data-driven completion of motion capture data," in *Proc. Workshop Virtual Reality Interact. Phys. Simulat. (VRIPHYS)*, Lyon, France, 2011, pp. 111–118.

[37] J. Xiao, Y. Feng, and W. Hu, "Predicting missing markers in human motion capture using *l*1-sparse representation," *Comput. Anim. Virtual Worlds*, vol. 22, nos. 2–3, pp. 221–228, 2011.

[38] R. Yan, L. Shao, and Y. Liu, "Nonlocal hierarchical dictionary learning using wavelets for image denoising," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4689–4698, Dec. 2013.

[39] J. Liu, X.-C. Tai, H. Huang, and Z. Huan, "A weighted dictionary learning model for denoising images corrupted by mixed noise," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 1108–1120, Mar. 2013.

[40] K. Engan, S. O. Aase, and J. H. Husøy, "Multi-frame compression: Theory and design," *Signal Process.*, vol. 80, no. 10, pp. 2121–2140, 2000.

[41] R. Ron, Z. Michael, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," Technion-Comput. Sci. Dept., Israel Inst. Technol., Haifa, Israel, Tech. Rep. CS-2008-08, 2008.

[42] L. N. Smith and M. Elad, "Improving dictionary learning: Multiple dictionary updates and coefficient reuse," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 79–82, Jan. 2013.

[43] Y. Feng, J. Xiao, Y. Zhuang, and X. Liu, "Adaptive unsupervised multi-view feature selection for visual concept recognition," in *Proc. 11th Asian Conf. Comput. Vis. (ACCV)*, Daejeon, Korea, 2012, pp. 343–357.

[44] H. Wang, F. Nie, W. Cai, and H. Huang, "Semi-superivised robust dictionary learning via efficient $\ell_{2,0^+}$-norms minimization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, 2013, pp. 1145–1152.

[45] H. Wang, F. Nie, and H. Huang, "Robust and discriminative self-taught learning," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, Atlanta, GA, USA, 2013, pp. 298–306.

[46] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 801–808.

[47] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1094–1121, Feb. 2012.

[48] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[49] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 600–616, Mar. 1997.

[50] E. T. Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence," *SIAM J. Optim.*, vol. 19, no. 3, pp. 1107–1130, 2008.

[51] L. Zhang *et al.*, "Discovering discriminative graphlets for aerial image categories recognition," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5071–5084, Dec. 2013.

[52] L. Zhang *et al.*, "Feature correlation hypergraph: Exploiting high-order potentials for multimodal recognition," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1408–1419, Oct. 2013.

[53] L. Zhang *et al.*, "Weakly supervised photo cropping," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 94–107, Jan. 2014.

[54] Y. Yang *et al.*, "A multimedia retrieval framework based on semi-supervised ranking and relevance feedback," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 723–742, Apr. 2012.

[55] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.

[56] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, Jun. 2014.

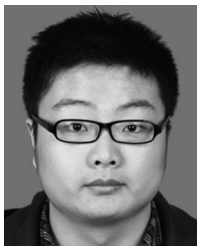
[57] Z. Ma, Y. Yang, N. Sebe, K. Zheng, and A. G. Hauptmann, "Multimedia event detection using a classifier-specific intermediate representation," *IEEE Trans. Multimedia*, vol. 15, no. 7, pp. 1628–1637, Nov. 2013.

[58] S. Becker, J. Bobin, and E. J. Candès, "NESTA: A fast and accurate first-order method for sparse recovery," *SIAM J. Imag. Sci.*, vol. 4, no. 1, pp. 1–39, 2011.

[59] R. Rubinstein, T. Faktor, and M. Elad, "K-SVD dictionary learning for the analysis sparse model," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Kyoto, Japan, 2012, pp. 5405–5408.

Yinfu Feng received the B.S. degree in information security from the University of Electronic Science and Technology of China, Chengdu, China, in 2009. He is currently pursuing the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China.

His current research interests include multimedia analysis and retrieval, computer vision, and machine learning.



Mingming Ji received the B.S. degree from the Computer School, Wuhan University, Wuhan, China, in 2008. He is currently pursuing the Master degree in computer science and technology from Zhejiang University, Hangzhou, China.

His current research interests include computer vision, computer animation, and 3-D model retrieval.

Jun Xiao received the Ph.D. degree in computer science and technology from the College of Computer Science, Zhejiang University (ZJU), Hangzhou, China, in 2007.

He is currently an Associate Professor with the Microsoft Visual Perception Laboratory, College of Computer Science, ZJU. His current research interests include computer animation, digital entertainment, and multimedia retrieval.

Xiaosong Yang received the bachelor's and master's degree in computer science from Zhejiang University, Hangzhou, China, in 1993 and 1996, respectively, and the Ph.D. degree in computing mechanics from the Dalian University of Technology, Dalian, China, in 2000.

He was a Post-Doctorate at the Department of Computer Science and Technology, Tsinghua University, from 2000 to 2002, and a Research Assistant at the Virtual Reality, Visualization and Imaging Research Centre, Chinese University of Hong Kong, Hong Kong, from 2001 to 2002. He is a Senior Lecturer with the National Centre for Computer Animation, Media School, Bournemouth University, Poole, U.K. His current research interests include 3-D modeling, animation, real-time rendering, virtual reality, virtual surgery simulation, and computer aided design.

Jian J. Zhang received the Ph.D. degree in Mechanical Engineering from Chongqing University, Chongqing, China, in 1987.

He is currently a Professor of Computer Graphics at the National Centre for Computer Animation, Bournemouth University, UK, where leads the Computer Animation Research Centre. He is also a cofounder of the UK's Centre for Digital Entertainment, Media School, Bournemouth University, Poole, U.K., which received an initial funding of over six million GBP from the Engineering and Physical Sciences Research Council. His research focuses on a number of topics relating to 3D virtual human modelling, animation and simulation, including geometric modelling, rigging and skinning, motion synthesis, deformation and physics-based simulation.

Yueting Zhuang received the Ph.D. degree in computer science from Zhejiang University (ZJU), Hangzhou, China, in 1998.

From 1997 to 1998, he was a Visitor at the Department of Computer Science and Beckman Institute, University of Illinois at Urbana-Champaign, Champaign, IL, USA. He is currently a Professor and the Vice Dean with the College of Computer Science, ZJU. His current research interests include computer animation, multimedia analysis, digital entertainment, and digital library technology.

Xuelong Li (M'02–SM'07–F'12) received the B.Eng. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China.

He is a Full Professor with the Center for OPtical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P. R. China.