Signal Processing 110 (2015) 108–122

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Sparse motion bases selection for human motion denoising

Jun Xiao [a], Yinfu Feng [a,*], Mingming Ji [a], Xiaosong Yang [b], Jian J. Zhang [b], Yueting Zhuang [a]

[a] School of Computer Science, Zhejiang University, Hangzhou 310027, PR China
[b] Bournemouth University, Poole Dorset BH12, United Kingdom

ARTICLE INFO

Article history:
Received 4 May 2014
Received in revised form
5 August 2014
Accepted 12 August 2014
Available online 20 August 2014

Keywords:
Human motion denoising
Data-driven
Poselet model
ℓ_1-minimization

ABSTRACT

Human motion denoising is an indispensable step of data preprocessing for many motion data based applications. In this paper, we propose a data-driven based human motion denoising method that sparsely selects the most correlated subset of motion bases for clean motion reconstruction. Meanwhile, it takes the statistic property of two common noises, i.e., Gaussian noise and outliers, into account in deriving the objective functions. In particular, our method firstly divides each human pose into five partitions termed as poselets to gain a much fine-grained pose representation. Then, these poselets are reorganized into multiple overlapped poselet groups using a lagged window moving across the entire motion sequence to preserve the embedded spatial–temporal motion patterns. Afterward, five compacted and representative motion dictionaries are constructed in parallel by means of fast K-SVD in the training phase; they are used to remove the noise and outliers from noisy motion sequences in the testing phase by solving ℓ_1-minimization problems. Extensive experiments show that our method outperforms its competitors. More importantly, compared with other data-driven based method, our method does not need to specifically choose the training data, it can be more easily applied to real-world applications.

© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).

1. Introduction

Motion capture is a powerful and mature technique for creating realistic computer character animation. It has been widely adopted in a large variety of applications such as animation production, computer games, human–computer interaction and medical rehabilitation [1–3]. In these applications, high-quality motion data are demanded for the purpose of accurate motion analysis and generation. However, even with a highly professional motion capture system there are many instances where some markers are occluded or mismatched [4–7]. As a result, it is necessary to fill the missing markers, while it may result in a certain percentage of noise. On the other hand, if some markers are mismatched when the tacking algorithm confuses the trajectory of one marker with that of another in some cases, the captured motion data contain serious error which can be regarded as bad noise or outliers. In order to clean the noisy data, most of the commercial motion capture systems provide various post process softwares for editing motion data including filling missing values and removing noise. To undertake the task of motion editing, the user must be patient and have professional knowledge of human motion capture. The underlying denoising/smoothing methods of these softwares mainly derive from linear and/or nonlinear interpretation methods, which suggests that they are only

* Corresponding author.
E-mail addresses: junx@zjuem.zju.edu.cn (J. Xiao), fyf200502@hotmail.com (Y. Feng), mjcs@zju.edu.cn (M. Ji), xyang@bournemouth.ac.uk (X. Yang), jzhang@bournemouth.ac.uk (J.J. Zhang), yzhuang@zju.edu.cn (Y. Zhuang).

http://dx.doi.org/10.1016/j.sigpro.2014.08.017
0165-1684 © 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).

efficient for dealing with simple or short-term noise cases. When these methods are used to handle some complex or long-term noise cases, the filtered motion will be distorted and unrealistic. That is to say, they may fail under these circumstances. Moreover, it is time-consuming and error-prone to process the noisy motion data in manual [8].

Meanwhile, more and more low-cost depth sensors (e.g., the Microsoft Kinect and SoftKinect) that can acquire the depth stream with acceptable accuracy have been released in recent years. With the aid of these new-fashioned products, many classic difficult computer vision problems like background subtraction and human detection become tractable. It also provides new opportunities for developing accessible motion capture. In light of this, some new algorithms have been proposed to recover human motion from the depth stream in real-time [9,10]. Compared with the traditional motion capture techniques such as the optical-based motion capture, the motion data generated from the depth stream are more likely to contain noise and outliers. For instance, if an actor performs the freestyle swimming action in front of a Microsoft Kinect, the recovered motion of the actor's two hands will be distorted due to the reason of self-occlusion of human body parts. In fact, researchers still have an uphill journey in improving the quality of these newly generated motion data.

To improve the aforementioned issues, a lot of researchers have plunged into the topic of motion data denoising over the years. Through a great deal of effort, a number of human motion denoising methods and techniques have been proposed. However, some intrinsic shortcomings of these methods hinder them from being widely applied in real-world applications. Take the popular signal-based denoising methods (e.g., Gaussian low-pass filter and discrete cosine transform (DCT)) for example, although they are easy to implement and only require a little of computational cost, they ignore the underlying structure correlation between different human joints and cannot preserve the embedded spatial–temporal motion patterns [11–15]. Indeed, human motion involves highly coordinated movement and the movement between different human joints are not independently [8]. As an improvement, the dynamic system based methods represented by Kalman filter and linear dynamic system (LDS) have been developed to discover hidden variables and learn their dynamics [16,17]. But a little of time delay will appear after motion denoising with the dynamic system based methods [18].

On the other hand, with the explosive growth of the available motion capture data in recent years, data-driven based motion denoising methods have attracted much attention [8,19]. Lou and Chai [8] proposed an example-based data-driven method to learn a series of filter bases, which hold some spatial–temporal patterns embedded in precaptured motion data, and then use them along with the robust statistics technique to filter noisy motion data [8]. Their method received encouraging results both on the real and simulated motion data. However, they use all of the learned filter bases to reconstruct the clean motion sequences indiscriminately, so their training database must be behavior-specific and typically only contains motion data selected from the same action with different style variants. Otherwise, the performance of their method will decline significantly since the bases learned from motion data with different action contain significantly different spatial–temporal patterns.

To overcome the shortcoming of [8], we propose a new data-driven based human motion denoising method in this paper. The key ideas of our paper are in twofold: sparsely selecting the most correlated subset of motion bases for clean motion reconstruction and taking the statistic property of motion noise into account in deriving our objective functions. The flowchart of our proposed method is illustrated in Fig. 1. And, the major contributions of our proposed method are summarized as follows.

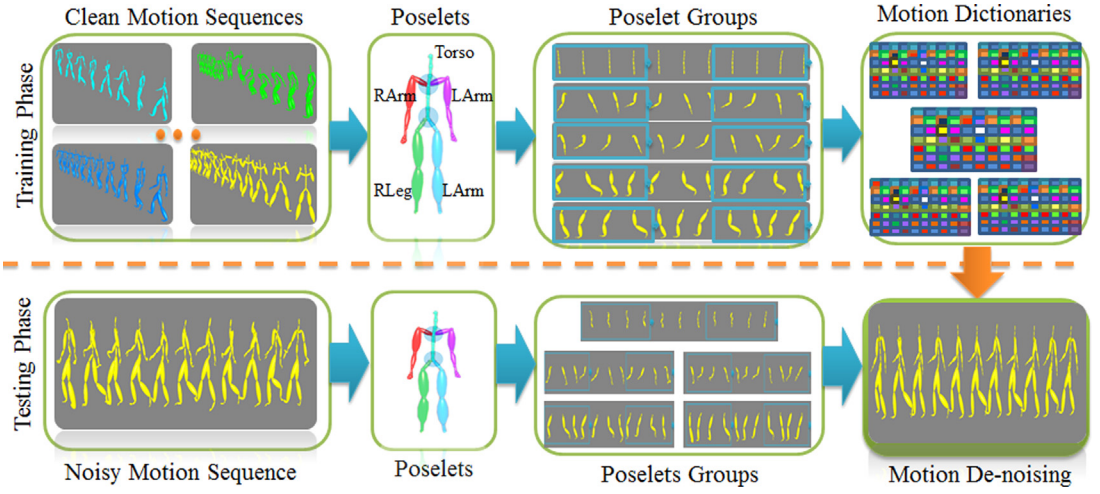

Fig. 1. The illustration of our proposed human motion data denoising framework. For the input motion sequences, we first divide each human pose into five partitions, which are termed as poselets. These poselets are then grouped together using a lagged window moving through the entire motion sequences to generate poselet groups. In the training phase, we use these poselet groups to learn five motion dictionaries and adopt these learned motion dictionaries to remove the noise and outliers from noisy poselet groups in the testing phase. Finally, we reorganize the filtered poselet groups to reconstruct the clean motion sequences.

1. A fine-grained human pose representation method termed as poselet model is proposed in this work. Using the entire human pose as a representation is a little coarser, and noisy data will inevitably influence the clean data. To avoid this issue, we divide a human pose into five parts and call them the poselets with a view to obtaining 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 representation does not only improves the performance of our algorithm, but also reduces the entire data processing time.
2. By utilizing the sparse sample selection ability of the ℓ_1-norm, we convert the classic human motion denoising problem into a ℓ_1-minimization framework. Different from the work [8], our method can automatically select the most correlated subset of motion bases from motion dictionaries, which are learned using the pre-captured motion data from either multiple different actions or just the same single type of action, for clean motion reconstruction. Thus, we do not need to specifically choose the training dataset. In other words, our method can be more easily applied to real-world applications.
3. For the two most common noises, i.e., Gaussian noise and outliers, two slightly different objective functions are proposed in this work by imposing the ℓ_2- and ℓ_1-norm constraints respectively. The former is optimal with respect to the Gaussian noise, while the latter is rather robust against outliers.

The structure of this paper is organized as follows. A review of related work is given in Section 2. The proposed human motion denoising method is described in Section 3, and experiments are shown in Section 4. Finally, in Section 5, concluding remarks are drawn and future research directions discussed.

2. Related work

In this section, we briefly review some related work in human motion denoising as well as dictionary learning that is involved in the training phase of our method.

2.1. Human motion denoising

Human motion denoising removes the noise and outliers while making the intrinsic information like the structure information of human body and the spatial–temporal patterns embedded in motion data is left intact. Over the last twenty years, a great deal of research effort has been done on this topic. Roughly speaking, the existing human motion denoising methods can be classified into three categories: signal-based methods, data-driven methods, and low-rank matrix based methods.

Since motion data can be regarded as a special multivariable signal, the traditional signal processing algorithms can be directly applied to handle the problem. Specially, these signal-based methods also can be divided into three subcategories as follows.

The first subcategory is to construct various motion filters from the perspective of signal filtering. For instance, the standard Gaussian low-pass, discrete cosine transform (DCT) and Fourier transform have been adopted in some earlier works [6,11,12,20–22]. Bruderlin and Williams [20] suggested that the techniques from the image and signal processing domain can be applied to design, modify, and adapt animated motion. Jehee and Shin [6] formulated filtering non-linear orientation data into a linear time-invariant filtering framework by transforming orientation data into a vector space and then transforming the results back to orientation space after applying filtering. Yamane and Nakamura [21] presented a dynamics filter that converts a physically inconsistent motion into a consistent one. In [22], the B-spline wavelet-based agent was proposed to remove impulsive noise embedded in noisy rigid body motion data. They decomposed the noisy data using multiresolution analysis. The noisy components are identified as coefficients of high magnitude. Consequently, the authors suggested to smooth these high-magnitude coefficients to remove the noise.

The second subcategory is to eliminate non-informative components of the signal by dimension reduction. This can be achieved using principal component analysis (PCA), which allows the expression of the original dataset in a new reduced subspace that maximizes its variance [23,24]. However, the low dimensions calculated by PCA only account for the variance of the data on some orthogonal directions. Sometimes, we are much more interested in a small subset of independent latent factors that contribute to generating different kinds of motion than the principal components. In other words, we hope to reveal such independent latent factors so that we can use them to reconstruct the clean motion. For this reason, independent component analysis (ICA) is another good choice, and it minimizes the statistical dependence of the representational components of motion data [25]. Later on, inspired by the great success of manifold learning in computer vision and machine learning [26–29], manifold learning methods also have been adopted in human motion denoising [30]. Indeed, they can be regarded as a special kind of dimensional reduction methods, which take the embedded manifold structure information of data into account.

The third subcategory is represented by linear dynamic system (LDS) and Kalman filter, which are applied to discover hidden variables and learn their dynamics [16,17,31]. Tak and Ko [32] converted a given captured or animated motion to a physically plausible motion by casting the motion editing problem as a constrained state estimation problem, based on the per-frame unscented Kalman filter framework. Shin and his colleagues [33] used a Kalman filter scheme to address motion capture noise issues in real-time computer puppetry situation. Wang et al. [34] proposed a latent variable model named Gaussian process dynamical models (GPDMs) to analyze nonlinear time-series data, such as the high-dimensional human motion capture data. They learned the GPDMs of human pose and motion from the captured human motion data, then applied them to remove the noisy data [34].

Usually, the above-mentioned signal-based methods are very fast and efficient in handling simple and short-term

noise cases. However, the structure information of human body has not been explicitly exploited, and the spatial–temporal patterns embedded in motion data also cannot be preserved by these methods.

With the explosive growth of the available motion capture data, data-driven based motion denoising methods have attracted much attention in recent years. For example, Lou and Chai [8] proposed an example-based data-driven method that first applies multi-channel singular spectrum analysis (M-SSA) to learn a series of filter bases, which hold some spatial–temporal patterns embedded in precaptured motion data, and then uses them along with robust statistics techniques to filter noisy motion data. Their method received perfect results both on real and simulated motion data. However, the shortcomings of their method are in twofold. First, only the top K (which is chosen by keeping 90% of the original motion energy) filter bases are kept and used in subsequent motion denoising phase, so it is unable to recover some motion details. In a mathematical sense, it is because that the remainder filter bases matrix is not a full rank matrix, and the bases cannot span the whole motion feature space. Second, they indiscriminately use all of the top K filter bases to reconstruct the clean motion, which requires that their training data must be carefully selected from the same action as that of the noisy data. Otherwise, the performance of their method will decline significantly, since filter bases learned from motion data with different action obviously contain different spatial–temporal patterns. Another example is that Akhter et al. [19] presented a bilinear spatio-temporal bases model by simultaneously exploiting spatial and temporal regularity while maintaining the ability to generalize well to new sequences. Their model can be interpreted as representing the data as a linear combination of spatio-temporal sequences consisting of shape models oscillating over time at key frequencies. They applied it to a number of analysis tasks including missing data filling and motion data denoising to demonstrate the effectiveness of their model. Compared with the other kind of motion denoising methods, the biggest advantage of data-driven methods is that they can automatically discover and learn some spatial–temporal patterns embedded in motion data. However, they may suffer from the out-of-sample problem, i.e., they cannot well handle the new 'unseen' human motions which have not been contained in the database.

In addition, Lai and Yuen [35] noticed that the approximately low-rank property of motion matrix has not been explicitly exploited, so they recasted the human motion data completion and denoising problems into a general low-rank matrix completion problem. Their proposed objective function is solved via the singular value thresholding (SVT) algorithm [36]. The key advantage of their method is that the above-mentioned out-of-sample problem is overcome, since their method does not need any training data. However, the user has to guess the standard deviation of the noise in their work, which is difficult in practice. Moreover, only imposing the low-rank structure property of human motion data in the objective function does not guarantee that the recovered human motion is smooth enough [18]. Indeed, the low-rank matrix completion theory would be failed to handle some badly corrupted human motion sequences.

2.2. Dictionary learning

As shown in Fig. 1, we need to learn five motion dictionaries, which contain spatial–temporal motion bases, in the training phase of our method. To facilitate the later discussions in the paper, we briefly review the methods of dictionary learning. Generally, the majority of works on the topic of dictionary learning can be broadly classified into three categories: example-based, analytical-based and learning-based.

The example-based methods directly use a lot of examples to construct an overcomplete dictionary. It is very simple and fast. And, this kind of method has been adopted in some applications like digit number recognition, human face classification [37] and missing markers prediction in human motion capture [38]. However, the constructed dictionaries are not compact. More importantly, the performance heavily rely on the selected examples.

The analytical-based methods construct the dictionary based on some pre-existing dictionaries like DCT bases and Wavelets bases [39]. These pre-existing bases are universal, and the obtained dictionaries are not task dependent. Besides, it is not trivial to decide how to choose these bases or modified them to make them fit different tasks.

The last learning-based approaches use machine learning techniques to infer the dictionary from a set of training examples. The main process of these methods can be divided into two stages: sparse coding and dictionary update [40–43]. Sparse coding [44–47] is to find the sparest solution of the training samples, while dictionary update runs when such a solution is found. These two stages iteratively run till that the algorithm is convergence. The most significant advantage of learning-based approaches over the other methods is that they can learn much finer and compact dictionaries than the other methods. MOD and K-SVD are two most popular dictionary learning methods which perform a critical role in many successful applications [48]. Though the learning-based methods often have to face the computational complexity problem, which limit the size of the trained dictionaries and the dimensions of the data that can be processed, some fast and large-scale variants of dictionary learning methods are available now [40,49–51]. In light of this, we choose a fast variant of K-SVD [40] to learn the motion dictionaries in the training phase in this work.

3. Sparse motion bases selection for human motion denoising

As mentioned above, the key ideas of our method are in twofold: sparsely selecting the most correlated subset of motion bases for clean motion reconstruction and taking the statistic property of noise into account in deriving the objective function. Based on these two ideas, we propose a sparse motion bases selection method for human motion denoising as illustrated in Fig. 1. In this section, we give the details of our proposed method as follows.

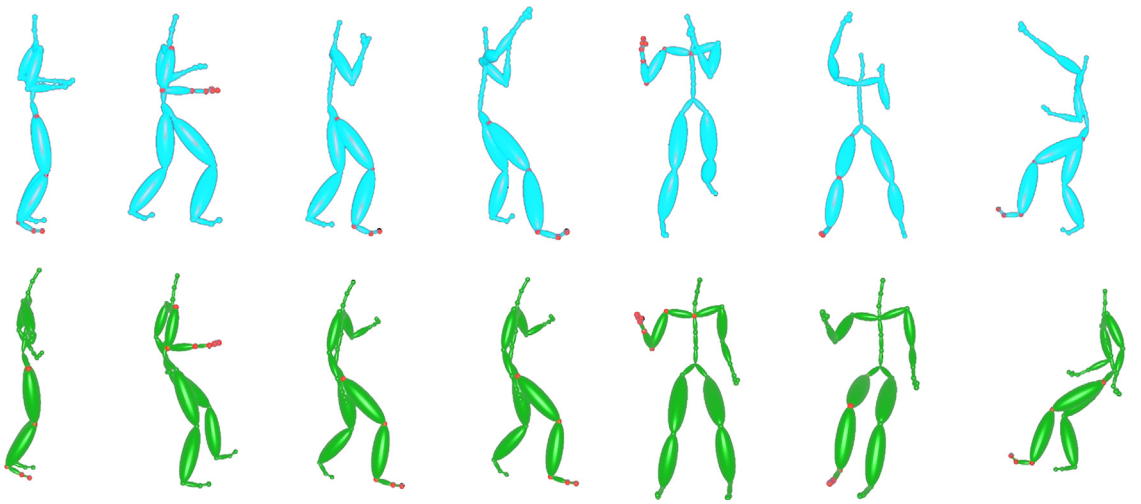

Fig. 2. Two different motion sequences from CMU motion database [61] with high local similarity at certain body parts whose markers are labeled with red color. In the top row, some poses from a boxing motion sequence are presented and the corresponding similar poses from a basketball motion sequence are given in the bottom row. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

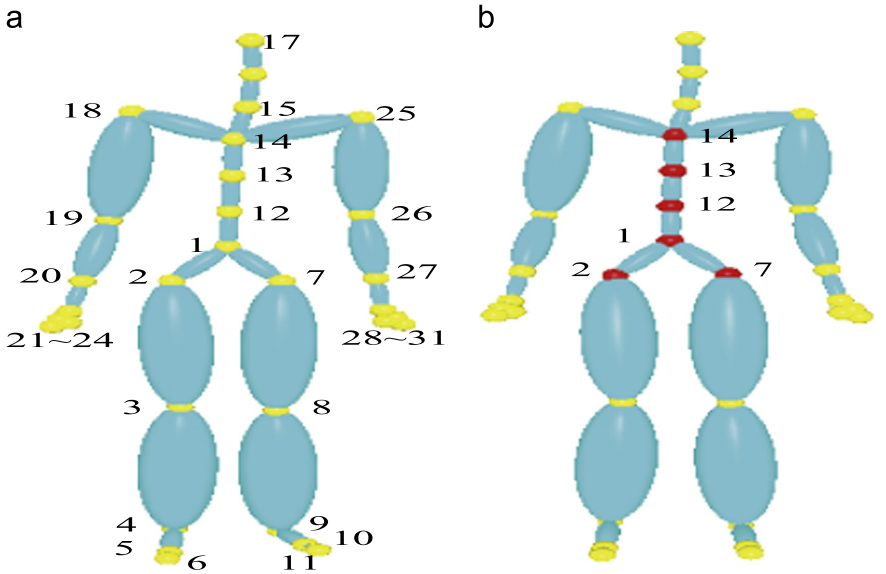

Fig. 3. The CMU pose model with 31 markers and the rigid part of human pose. The markers belong to the rigid part (i.e., marker-1,2,7,12,13,14) are labeled with red color. The markers with numbers 1, 2, 7 and 14 are the root, the right and left femur markers, and the upper neck marker, separately. (a) CMU pose model and (b) rigid part of pose. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

3.1. Preprocess

3.1.1. Coordinate translation

Although the real-world global coordinates of the human motion are of highly complexity and variance, their local coordinates with respect to the root marker are inactive (e.g., the torso of the human body in a walking motion) and frequently contain body parts with similar postures and movements. Even if motion sequences contain different actions, they also can share some similar body part postures and local spatial–temporal patterns in the local coordinate system as shown in Fig. 2. Based on this observation, we try to reveal and exploit these local similarity information for human motion denoising. Meanwhile, we notice that the torso as illustrated in Fig. 3(b) is a stable structure wherein the distance between every two markers is nearly constant and can be mostly regarded as a rigid part. Therefore, we normalize each pose and then translate the original global motion into the local motion according to joints belonging to the rigid part. In particular, we first translate each normalized pose [52] to make its root marker in the origin of the local coordinate system. Then, we rotate the local pose to make the plane consisting of three markers, i.e., the left femur, the right femur and the upper neck, parallel with the *XY*-plane. In addition, the ray that passes through the middle point between the left and the right femur markers and the upper neck marker is also parallel with the *Y*-axis and

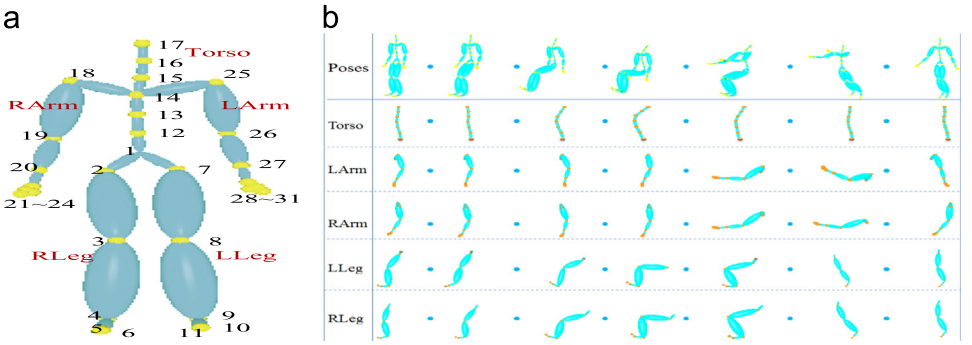

Fig. 4. The illustration of poselet generation. In the left subfigure, a human pose is divided into five parts: Torso, Left Arm (LArm), Right Arm (RArm), Left Leg (LLeg) and Right Leg (RLeg), which are termed as poselets. The subsets of markers in these five parts are $\{1,12,13,14,15,16,17\}$, $\{14,25,26,27,28,29,30,31\}$, $\{14,18,19,20,21,22,23,24\}$, $\{1,7,8,9,10,11\}$ and $\{1,2,3,4,5,6\}$ respectively. The joint markers like marker-14 and marker-1 can belong to multiple poselets in order to make the filtered results stable. The right subfigure presents some poselets in a jump motion sequence. (a) Pose and poselet model and (b) poselets in a motion sequence.

points to the positive direction of the Y-axis. Note that we record all of these transformation information into a matrix $M = M_r \times M_t$ wherein M_t is the translation matrix and M_r is the rotation matrix. All of the operations can be reversed, which means that after motion denoising we can convert such local poses back into the global poses.

3.1.2. Poselet generation

Suppose a normalized local motion sequence consists of T poses and each pose contains L markers, we denote it as $\mathbf{X} = [\mathbf{p}_1, \ldots, \mathbf{p}_T]$, where $\mathbf{p}_t = [x_{t,1}, y_{t,1}, z_{t,1}, \ldots, x_{t,L}, y_{t,L}, z_{t,L}]^T$ represents the t-th pose/frame.

Since using all of the markers as the pose feature representation is a little coarser, we divide each human pose into five parts, which are termed as poselets, to obtain a much more fine-grained pose representation. The five poselets are Torso (contains head), left arm (LArm), right arm (RArm), left leg (LLeg) and right leg (RLeg), each of them is a set of markers as shown in Fig. 4. To make the position of the joint markers like marker-1 and marker-14 stable, we assign them to multiple poselets as shown in Fig. 4(a).

For each poselet, one submatrix is derived from $\mathbf{X}$ and we denote the i-th poselet as $\mathbf{X}^i = [\mathbf{p}_1^i, \ldots, \mathbf{p}_T^i] \in \mathbb{R}^{d_i \times T}$, $i = 1, \ldots, 5$, where $\mathbf{p}_t^i$ just includes the subset of markers of $\mathbf{p}_t$ in the i-th poselet and d_i is the feature dimension of $\mathbf{p}_t^i$. Indeed, d_i equals to the number of markers in the subset timing three. With this kind of pose representation, we can speedup human motion denoising via processing these five poselets in a parallel manner.

3.1.3. Poselet grouping by a lagged window

If we process each human pose[1] one by one, the embedded spatial–temporal 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. Similar to [8], we adopt a lagged window with the length of M-frames moving across the entire motion sequence as shown in Fig. 1 and group all of the poselets in a same window into a group. The above obtained poselets are reorganized into $T-M+1$ overlapped groups. We reshape each group into a vector $\mathbf{g}_j^i = \Omega([\mathbf{p}_j^i, \mathbf{p}_{j+1}^i, \ldots, \mathbf{p}_{j+M-1}^i]) \in \mathbb{R}^{(d_i \times M) \times 1}$, where $j = 1, \ldots, T-M+1$ and Ω is defined as the vectorization operation that reshapes a matrix into a vector by stacking all columns one by one. We use $\mathbf{g}_j^i$ as the motion denoising processing primitive. And, we can totally derive five group motion matrices, i.e., $\mathbf{Y}^i = [\mathbf{g}_1^i, \ldots, \mathbf{g}_S^i], S = T-M+1$, from $X^i, i = 1, \ldots, 5$ by poselets grouping operation.

3.2. Motion dictionary learning using K-SVD

A natural human motion sequence contains some special spatial–temporal patterns. In order to reveal these patterns, we resort to the dictionary learning method and use multiple clean motion sequences to construct the five group motion matrices $\mathbf{B}^i = [\mathbf{Y}_1^i, \ldots, \mathbf{Y}_q^i] \in \mathbb{R}^{(d_i \times M) \times N}, i = 1, \ldots, 5$ according to Section 3.1.3. Here $\mathbf{B}^i$ can consists of either multiple different kinds of human motion sequences or just the same kind of motion sequences as that of the noisy motion. Then, we minimize the following optimization problem:

$$\min_{\mathbf{D}^i, \mathbf{W}^i} \quad \|\mathbf{B}^i - \mathbf{D}^i \mathbf{W}^i\|_F^2$$
$$\text{s.t.} \quad \mathbf{W}^i = [W_1^i, \ldots, W_N^i], \quad \|W_j^i\|_0 \le T_s, \quad \forall\ i,\ 1 \le j \le N \tag{1}$$

to obtain five corresponding motion dictionary matrices, i.e., $\mathbf{D}^i \in \mathbb{R}^{(d_i \times M) \times K_i}, i = 1, \ldots, 5$ where K_i is the column number of $\mathbf{D}^i$ and will be discussed in more detail later. In this formulation, $\mathbf{W}^i$ is the representation coefficient matrix, T_s is the target sparsity and $\|\cdot\|_0$ is the ℓ_0 pseudo-norm which counts the non-zero entries. In each motion dictionary matrix, e.g. $\mathbf{D}^i$, its columns are the desired motion bases which preserve the embedded spatial–temporal patterns in the group motion matrix $\mathbf{Y}^i$. For simplicity, we assume hereon that the columns of $\mathbf{D}^i$ are normalized to unit ℓ_2-length.

Eq. (1) is actually a non-convex problem with respect to $\mathbf{D}^i$ and $\mathbf{W}^i$. However, when we fix $\mathbf{D}^i$, the above sparsity-constrained problem (i.e., Eq. (1)), which is known to be NP-hard, can be relaxed and approximately solved using several available approximation techniques, including Orthogonal Matching Pursuit (OMP) [53], Basis Pursuit (BP) [54] and FOCUSS [55] and so on. Alternatively, when

[1] Note that each pose is just one frame of a motion sequence.

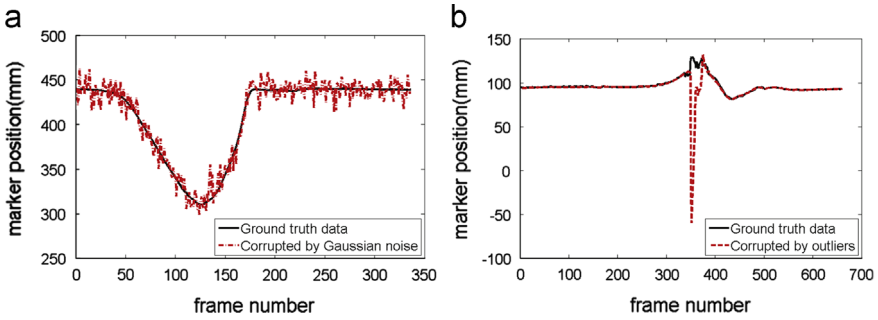

Fig. 5. The comparison of Gaussian noise and outlier. (a) Motion data corrupted by Gaussian noise and (b) motion data corrupted by outlier.

$\mathbf{W}^i$ is fixed, Eq. (1) becomes a convex problem and has a closed-form solution. Thus, using dictionary learning method to solve this problem can be divided into two stages: sparse coding and dictionary update. Sparse coding is to find the sparest solution of the training samples, while dictionary update runs when such a solution is found. There two stages iteratively run, until the algorithm is convergence. That is to say, a fundamental question in solving Eq. (1) is the choice of how to set or update the dictionary $\mathbf{D}^i$.

There exist several efficient dictionary learning methods such as the classic MOD, K-SVD, which can be used to solve Eq. (1). We choose to use a fast variant of K-SVD [40] in our work. K-SVD is a highly effective method of training overcomplete dictionaries for sparse signal representation and has successfully applied in various applications. Formally, K-SVD aims to iteratively improve the dictionary to achieve sparser representations of the signals in the data matrix $\mathbf{B}$, where a set of training samples are arranged as its columns, by solving the optimization problem

$$\min_{\mathbf{D},\mathbf{W}} \quad \|\mathbf{B}-\mathbf{D}\mathbf{W}\|_F^2$$

$$\text{s.t.} \quad \mathbf{W}=[W_1,\ldots,W_N], \quad \forall\ i \quad \|W_j\|_0 \le T_s, \quad 1\le j\le N. \tag{2}$$

The update of the dictionary columns is combined with an update of the sparse representations, thereby accelerating convergence. Besides, the K-SVD algorithm is flexible and can work with any pursuit method (e.g., OMP [53], BP [54], FOCUSS [55]).

However, the classical K-SVD algorithm is quite computationally demanding, especially when the dimensions of the dictionary are high or the number of training data becomes large. To overcome this problem, a fast implementation of K-SVD using batch orthogonal matching pursuit method reported in the work [40]. We use the Matlab toolbox[2] provided by the authors [40] in our experiments. Solving Eq. (1), we can get the five motion dictionary matrices, i.e., $\mathbf{D}^i$, $i=1,\ldots,5$, which will be used for human motion denoising.

[2] http://www.cs.technion.ac.il/~ronrubin/software.html

3.3. Motion denoising via ℓ_1-minimization framework

For each input motion sequence, we can generate its poselets and poselet groups using the above described poselet generation and grouping techniques. A significant difference between the testing phase (or the human motion denoising phase) and the training phase is that the captured motion sequences in the testing phase often contain the noise and outliers. In order to remove the noise and outliers, we propose a data-driven based human motion denoising method with the aid of the previously learned five motion dictionary matrices, i.e. $\mathbf{D}^i, i=1,\ldots,5$ and reformulate the motion denoising problem into a general ℓ_1-minimization framework.

In practice, Gaussian noise and outliers are two most common types of noises in human motion data. Thus, we take their statistic into account in human motion denoising. As shown in Fig. 5(a), Gaussian noise contaminates motion data by making marker position wavily drift from original point through the whole motion. However, outliers usually last only a few frames and behaves like a pulse signal as shown in Fig. 5(b).

Suppose the i-th noisy group motion matrix and the corresponding clean one are denoted as $\mathbf{Z}^i$ and $\mathbf{Y}^i$, the contained noise can be represented as $\mathbf{E}^i=\mathbf{Z}^i-\mathbf{Y}^i$. As mentioned above, we hope to select the most correlated subset of motion bases to reconstruct the clean motion data so that we do not need to specially select behavior-specific motion data, which come from the same action as the noisy motion data, with different style variants. In other words, our method can learn motion dictionaries from motion data with different actions and automatically select a most correlated subset of motion bases to reconstruct the clean motion. To achieve this goal, we optimize the following ℓ_1-minimization objective function:

$$\min_{\boldsymbol{\theta}^i}\|\mathbf{Z}^i-\mathbf{D}^i\boldsymbol{\theta}^i\|_p^p+\lambda\|\boldsymbol{\theta}^i\|_1 \tag{3}$$

where $p\in\{1,2\}$ and λ is a sparse regularized parameter. For a matrix X, $\|X\|_1=\sum_{i,j}|X_{i,j}|$.

For the Gaussian noise, the least square regression is the optimal method to filter it [56,57]. Thus squared ℓ_2-norm should be chosen for the above ℓ_1-minimization problem and it leads to

$$\min_{\boldsymbol{\theta}^i}\|\mathbf{Z}^i-\mathbf{D}^i\boldsymbol{\theta}^i\|_2^2+\lambda\|\boldsymbol{\theta}^i\|_1. \tag{4}$$

Eq. (4) is a ℓ_2/ℓ_1 denoising model, which can be solved by quadratic programming.

For the outlier, the previous ℓ_1-regularized least square regression method may fail, because the ℓ_2-norm tends to severely penalize the outliers and propagate the residual in the objective function uniformly. To get around this problem, we modify our denoising method by replacing the ℓ_2-norm with the ℓ_1-norm when outliers exist [58,57]. As pointed out previously, because outlier noise is usually very sparse, the ℓ_1-norm is preferred. As a result, for outliers, our objective function becomes

$$\min_{\boldsymbol{\theta}^i} \|\mathbf{Z}^i - \mathbf{D}^i\boldsymbol{\theta}^i\|_1 + \lambda\|\boldsymbol{\theta}^i\|_1. \tag{5}$$

Eq. (5) is a ℓ_1/ℓ_1 denoising model, which can be solved using the alternating direction algorithm [59].

It is necessary to mention that since we have proposed two slightly different objective functions to deal with Gaussian noise and outliers separately, if we know which kind of noise is dominant in the motion data in advance, we can choose the corresponding denoising model very easily, although in practice, it is difficult to have enough prior knowledge about the noise. However, we found that combining these two denoising models and filtering noisy motion data one by one received encouraging results in the experiments. Therefore we can always filter it with the ℓ_1/ℓ_1 denoising model first to remove the outliers and then refine the result with the ℓ_2/ℓ_1 denoising model, which removes some remainder Gaussian noise.

After solving Eqs. (4) and (5), we get the sparse reconstruction coefficient matrix $\mathbf{W}^i, i=1,\ldots,5$. Then, we can reconstruct the filtered clean group motion matrix via $\tilde{\mathbf{Y}}^i = \mathbf{D}^i\boldsymbol{\theta}^i$. Recall that $\mathbf{Y}^i = [\mathbf{g}_1^i,\ldots,\mathbf{g}_S^i], S = T-M+1$ and $\mathbf{g}_j^i = \Omega([\mathbf{p}_j^i, \mathbf{p}_{j+1}^i, \ldots, \mathbf{p}_{j+M-1}^i]) \in \mathbb{R}^{(d_i\times M)\times 1}$. So, we decompose the filtered poselets groups $\tilde{\mathbf{g}}_j^i$ in $\tilde{\mathbf{Y}}^i$ and calculate the mean value for each poselet, e.g. $\tilde{\mathbf{p}}_j^i = (1/n)\sum_{t=1}^{n}(\tilde{\mathbf{p}}_j^i)_t$ wherein $(\tilde{\mathbf{p}}_j^i)_t$ is the t-th copy of $\tilde{\mathbf{p}}_j^i$ and n is the total number of copy of the poselet $\tilde{\mathbf{p}}_j^i$ in $\tilde{\mathbf{Y}}^i$. Since the i-th poselet is $\mathbf{X}^i = [\mathbf{p}_1^i,\ldots,\mathbf{p}_T^i] \in \mathbb{R}^{d_i\times T}, i=1,\ldots,5$, we can recover the filtered submatrix $\tilde{\mathbf{X}}^i$ based on the recovered poselet $\tilde{\mathbf{p}}_j^i$. It is also easy to form the local motion matrix $\tilde{X}$. Finally, we translate the local poses to be the global poses according to the recorded transformation matrix M in the process of coordinate translation. The whole flowchart of our proposed method is illustrated as shown in Fig. 1. Meanwhile, we summarize the algorithm of our method in Algorithm 1.

Algorithm 1. Sparse motion bases selection for human motion denoising.

Input: motion matrices:$\mathbf{D}^i, i=1,\ldots,5$; the input global noisy motion sequence:X_{global}; the length of the moving window:M; the regularized parameter: λ.

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

1: **Coordinate Translation**:
change the global noisy motion sequence X_{global} into the local noisy motion sequence X_{local};

2: **Poselet Generation**:
generate poselets $\mathbf{X}^i = [\mathbf{p}_1^i,\ldots,\mathbf{p}_T^i] \in \mathbb{R}^{d_i\times T}, i=1,\ldots,5$ based on X_{local}

3: **Poselet Grouping**:
generate poselet groups $\mathbf{Y}^i = [\mathbf{g}_1^i,\ldots,\mathbf{g}_S^i], S = T-M+1$ according to $X^i, i=1,\ldots,5$

4: **Motion Denoising**:
if(Gaussian noise)
apply the ℓ_2/ℓ_1 denoising model and solve Eq. (4) to obtain the filtered $\tilde{\mathbf{Y}}^i, i=1,\ldots,5$;
elseif (outlier)
apply the ℓ_1/ℓ_1 denoising model and solve Eq. (5) to obtain the filtered $\tilde{\mathbf{Y}}^i, i=1,\ldots,5$;
otherwise(mixed noise)
combine the ℓ_1/ℓ_1 and ℓ_2/ℓ_1 denoising models to obtain the filtered $\tilde{\mathbf{Y}}^i, i=1,\ldots,5$;

5: **Decompose Poselet Groups**:
decompose poselet groups $\tilde{\mathbf{Y}}^i, i=1,\ldots,5$ to obtain multiple poselets.

6: **Calculate Filtered Poselets**:
calculate the mean value for each poselet, e.g. $\tilde{\mathbf{p}}_j^i = \frac{1}{n}\sum_{t=1}^{n}(\tilde{\mathbf{p}}_j^i)_t$ wherein $(\tilde{\mathbf{p}}_j^i)_t$ is the t-th copy of $\tilde{\mathbf{p}}_j^i$ and n is the total number of copy of the poselet $\tilde{\mathbf{p}}_j^i$ in $\tilde{\mathbf{Y}}^i$.

7: **Form Local Motion Matrix**:
form the filtered submatrix $\tilde{\mathbf{X}}^i$ based on the recovered poselet $\tilde{\mathbf{p}}_j^i$ and then obtain the filtered local motion matrix $\tilde{X}_{local}$.

8: **Coordinate Translation**:
change the local filtered motion sequence $\tilde{X}_{local}$ into the global filtered motion sequence $\tilde{X}_{global}$;

3.4. Time complexity analysis

The computational cost of our proposed method mainly comes from two steps: one is to learn the motion dictionaries via solving Eq. (2) and the other is to denoise the noisy motion data via solving Eq. (4) (i.e., the ℓ_2/ℓ_1 denoising model) or Eq. (5) (i.e., the ℓ_1/ℓ_1 denoising model). As we have seen, it is possible to implement our proposed method in parallel mode, so we just need to run these two steps for each poselet only one time. Moreover, it has been proofed that Eq. (5) can be reformulated into the same form as Eq. (4) in [59]. Thus, the time complexity of the proposed method is calculated from two parts:

- The calculation of $\mathbf{D}^i$ via the fast K-SVD [40]: the time complexity is $O(N((T_s)^2+2d_i\times M)\times K_i)$, where N is the number of training poselet groups in $\mathbf{B}^i$ of Eq. (1).
- The calculation of $\boldsymbol{\theta}^i$ via solving Eq. (4) or Eq. (5): for the ℓ_1 minimization problem, we adopt the dual augmented Lagrangian method (DALM) [60] to calculate $\boldsymbol{\theta}^i$. The time complexity of this part is $O(S(d_i\times M)(d_i\times M+K_i))$.

Assuming an asymptotic behavior of $T_s < K_i < N$ and $d_i\times M < K_i$, the first part of computational cost simplifies to the following expression: $O(N\times 2\times d_i\times M\times K_i)$. Similarly, the second part becomes $O(S\times d_i\times M\times K_i)$. Therefore, the entire time complexity of our method is about $O((2N+S)\times d_i\times M\times K_i)$. In practice, S dependents on the input noisy motion sequences and most of the motion sequences in CMU motion dataset [61] belong to short-length sequences. Even the long-length motion sequences, we can first segment them into multiple short-length motion sequences and then filter each short-length

motion sequences in parallel. So, S is usually less than N. In other words, most of our computational cost is expended on training motion dictionaries using the fast K-SVD method. Fortunately, we have found that it just spent no more than 10 min to train the motion dictionaries in all of our experiments. Thus, our method can be widely used in real-world applications.

4. Experiments

4.1. Experimental setup

Since the quality of the filtered motion would be effected by different cues like the category of motion, the noise level and type, we compare the proposed method with other methods under various conditions. We choose three representative activities, i.e., run, dance and basketball from CMU human motion database [61] for our experiments.[3] It is because that run is a common simple activity, which contains many repetitive movements while dancing and basketball are two more complex sports that contain a few of repetitive movements. Besides, the length of these motion sequences is slightly different, so the experimental dataset includes short, middle and long human motion sequences. For each activity, we collect 20 sequences of motion data from more than 2 subjects. We then randomly select 80 percentage of data as the training data while the remainder of each action is used as the testing data. Since most of the CMU human motion data are very clean, we use the training data to learn dictionary matrices for our method and [8]. For the testing data, we automatically synthesize three kinds of noise: (1) Gaussian noise with the signal-to-noise ratio (often abbreviated SNR) ranges from $\{40, 30, 20, 10\}$ dB; (2) outlier with the ratio from 10% to 25% with an interval of 5%; (3) mixed noise that consists of both Gaussian noise (SNR=20 dB) and outliers (ratio=10%), for each motion sequence. The outliers are generated by multiplying 1.4 to the selected entries in motion data matrix.

We quantitatively assess the performance of our method by comparing it with other four widely used human motion denoising methods, i.e., Gaussian filter, Wavelet filter [5,22], Kalman filter [33,32] and the example-based method [8]. For the former three, we apply them to remove the noise and outliers from each feature dimension of motion data independently. To make a fair comparison, we tune parameters for each algorithm by the way of cross-validation using the training data and report their best results. For instance, we tune the size of lagged window from $\{11, 21, 31, 41, 51\}$ and the parameter of Welsch estimator (i.e., p) from $\{3, 5, 7, 9\}$ for the example-based method [8] following the work [8]. Though the authors suggested to determine the number of reserved bases K by keeping 99% of the original motion variation, we have observed that K will be too small (less than 5) by setting it in such a way, due to the reason that motion data are usually approximated low-rank [35]. So, we tune K from $\{20, 40, 60, 80, 100\}$ for the example-based method and choose the best setting. Similarly, we tune the size of the lagged window from $\{9, 15, 21, 27, 33, 39\}$ for our method. Since our method is a sparse-based method, the motion dictionary matrices $\mathbf{D}^i \in \mathbb{R}^{(d_i \times M) \times K_i}, i = 1, \ldots, 5$, are flat shape matrices (in fact, they are overcomplete matrices), which means that $K_i \geqslant (d_i \times M)$. Thus, we tune K_i from 500 to 1000 with an interval of 50. For simplicity in the experiments, we set $K_i, i = 1, \ldots, 5$, to be the same value and denote it as K to be consist with the work [8]. And, the sparse regularized parameter λ is tuned from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$.

4.2. Experimental results

To quantify the filtered results, following the work [17,38,52], the Root Mean Squared Error (RMSE) measurement is adopted:

$$RMSE(\mathbf{p}_i, \hat{\mathbf{p}}_i) = \sqrt{\frac{1}{n_e}\|\mathbf{p}_i - \hat{\mathbf{p}}_i\|^2}, \tag{6}$$

where $\mathbf{p}_i$ is the noisy pose, $\hat{\mathbf{p}}_i$ is the filtered clean pose and n_e is the total number of noisy markers in $\mathbf{p}_i$. Due to the limited space here, in order to facilitate the discussion, for each activity, we only present the results from only one sequence of the motion.[4]

The performance comparison results on three kinds of noise are shown in Figs. 6 and 7 and Table 1. From these results, we get the following conclusions:

1. Our proposed method outperforms the other competitors. More importantly, the variances of RMSE of our method are smaller than the others', which means that the outputs of our method are much stable than the others', as shown in Figs. 6 and 7 and Table 1. We believe it is owing to: (a) the proposed poselet model is a much fine-grained representation; (b) the ℓ_1-minimization framework takes both motion bases selection and the statistic property of noise into account.
2. Our method, the example-based method and Wavelet based method are the top three methods in most of the time as shown in Figs. 6 and 7 and Table 1.
3. When the added noise is just Gaussian noise, both our method and the example-based method [8] can work very well if the value of SNR is bigger than 20 dB. However, if the motion data are badly corrupted as shown in Fig. 6(d), (h) and (i), the outputs of all algorithms become a little less stable. In other words, it becomes much difficult to recover the clean motion under such bad condition.
4. When the added noise is outlier, the curves of all algorithms are less stable than their counterparts under the Gaussian noise condition. The denoising motion data of all algorithms are easy to contain some short time maker shakes which lead to some peaks appear in

[3] In this work, we use motion data converted from the asf/amc motion files provided in CMU motion capture database [61]. Each human pose contains 31 markers.

[4] The selected motion sequences are 09_04 (run), 05_16 (dance), and 06_13 (basketball).

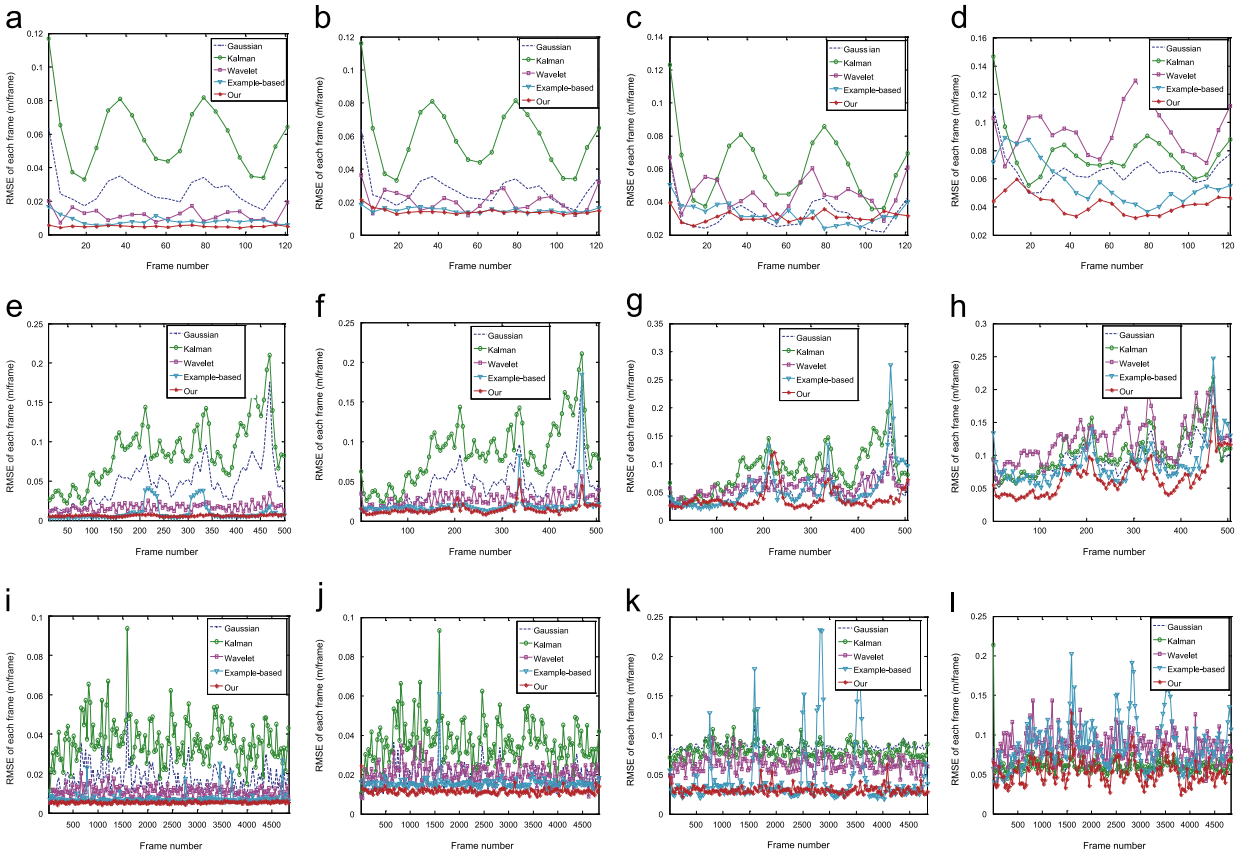

Fig. 6. Motion denoising comparisons of different algorithms on three human motion sequences with different degrees of Gaussian noise. The smaller the value of SNR, the heavier the added Gaussian noise. (a) Run (SNR = 40 dB), (b) Run (SNR = 30 dB), (c) Run (SNR = 20 dB), (d) Run (SNR = 10 dB), (e) Dance (SNR = 40 dB), (f) Dance (SNR = 30 dB), (g) Dance (SNR = 20 dB), (h) Dance (SNR = 10 dB), (i) Basketball (SNR = 40 dB), (j) Basketball (SNR = 30 dB), (k) Basketball (SNR = 20 dB) and (l) Basketball (SNR = 10 dB).

these curves. Since the value of RMSE of our method is usually less than 0.05 m/frame when the ratio of outlier is less than 20%, the recovered motion is visually acceptable.

5. When the added noise is mixed noise, which contains both Gaussian noise and outliers, our method and the example-based method [8] outperform the others. Meanwhile, the outputs of our method are much stable than that of [8], because the standard deviations of our method are much smaller than its competitor's [8] as shown in Table 1.

Based on the above experimental results, our method, the example-based method and wavelet-based method are the top three methods in most of the time, we present some recovered key poses of these methods on the three motion sequences under the mixed noise condition in Fig. 8 wherein the markers with a large deviation from its original location (>8 cm) are marked with yellow circles. From Fig. 8, we can see that most of the recovered key poses are visually acceptable and the filtered key poses of our method are close to the clean ones.

To demonstrate the benefit of motion bases selection, we use the motion data with Gaussian noise (SNR = 30 dB) as experimental data. We compare the performance variance of the two data-driven based methods (i.e., our method and the example-based method) using single motion, where motion data are selected from the same action category as the noise motion, or multiple motion data, where motion data are selected from multiple actions include the same action as the noise motion, as the training data. As shown in Fig. 9, the performance of our method can be improved using multiple motion data as the training data, while its competitor's performance is dropped. This is because that our method can optimally select the most correlated subset of motion bases for motion reconstruction with the aid of ℓ_1-norm, while the example-based method just uses all of its motion bases which may contain some irrelevant motion bases when using multiple motion as its training data. Additionally, our method use a much fine-grained motion representation method, i.e., poselet model as well as the poselet grouping techniques. Therefore similar body parts and motion patterns in different motion can be well exploit to improve the performance. To verify this idea, we compare both the running time and the mean RMSE of our method using the poselet representation and the traditional pose representation. As mentioned above, since the poselet representation brings in the potential possibility that we can process each poselet in a parallel mode, we have implemented our algorithm in parallel computation. The experiment is conducted

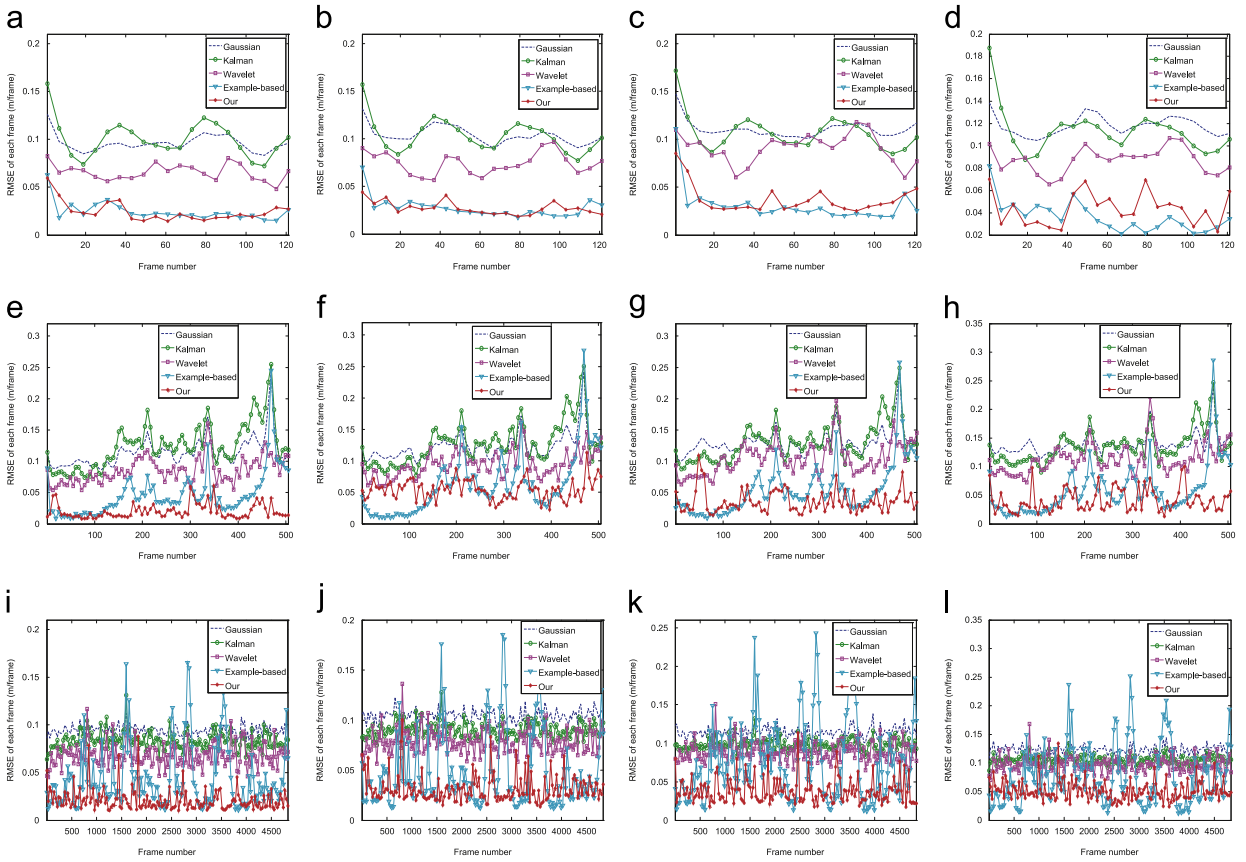

Fig. 7. Motion denoising comparisons of different algorithms on three human motion sequences with different ratio of outliers. The higher the value of Ratio, the heavier the added outliers. (a) Run (Ratio = 10%), (b) Run (Ratio = 15%), (c) Run (Ratio = 20%), (d) Run (Ratio = 25%), (e) Dance (Ratio = 10%), (f) Dance (Ratio = 15%), (g) Dance (Ratio = 20%), (h) Dance (Ratio = 25%), (i) Basketball (Ratio = 10%), (j) Basketball (Ratio = 15%), (k) Basketball (Ratio = 20%), (l) Basketball (Ratio = 25%).

Table 1
Comparisons of our method with other motion denoising algorithms on three human motion sequences with mixed noise. The average RMSE values of each frame (cm/frame) and standard deviations are reported. The best performance is highlighted in each case.

Action	Gaussian	Kalman	Wavelet	Example-based	Our
Run	10.25 ± 0.81	10.52 ± 1.95	7.92 ± 1.06	4.66 ± 2.27	$\mathbf{4.23 \pm 0.65}$
Dance	11.98 ± 2.26	12.74 ± 3.54	9.40 ± 1.94	6.85 ± 4.27	$\mathbf{6.27 \pm 3.28}$
Basketball	10.03 ± 0.59	8.89 ± 2.34	7.84 ± 1.13	5.51 ± 3.46	$\mathbf{4.30 \pm 1.16}$

on an Intel Xeon X5650 workstation at 2.66 GHz, using the MATLAB language to implement all the codes. As shown in Table 2, we find that the poselet representation reduces the running time and improves the performance of our method.

We also conduct experiments using the same three motion sequences to study how the different parameters affect the denoising performance. In Fig. 10, we report the performance variation of our method with respect to the number of poselets in a group (it is denoted as M), the size of dictionary K and the regularization parameter λ. Note that we set K is the same for all of the five learned motion dictionaries in this paper. From Fig. 10(a) and (b), we find that the bigger the value of N and K, the better the performance of our method. However, it also needs much more time to solve the objective function and more clean examples for the training. From Fig. 10(c), we find the optimal value of λ is 100, which ensures $\mathbf{W}^i$ is sparse. In other words, our method automatically selects a few most related motion bases for motion reconstruction.

5. Discussion and conclusion

In this paper, we have presented a data-driven based human motion denoising method that sparsely selects the most correlated subset of motion bases for clean motion reconstruction. It takes the statistic property of noise into account in deriving our objective function. A fine-grained human motion representation method called the poselet model was proposed; the poselet generation and grouping techniques were also adopted to reveal the embedded spatial–temporal patterns in human motion data. The classic human motion denoising problem was rewritten into

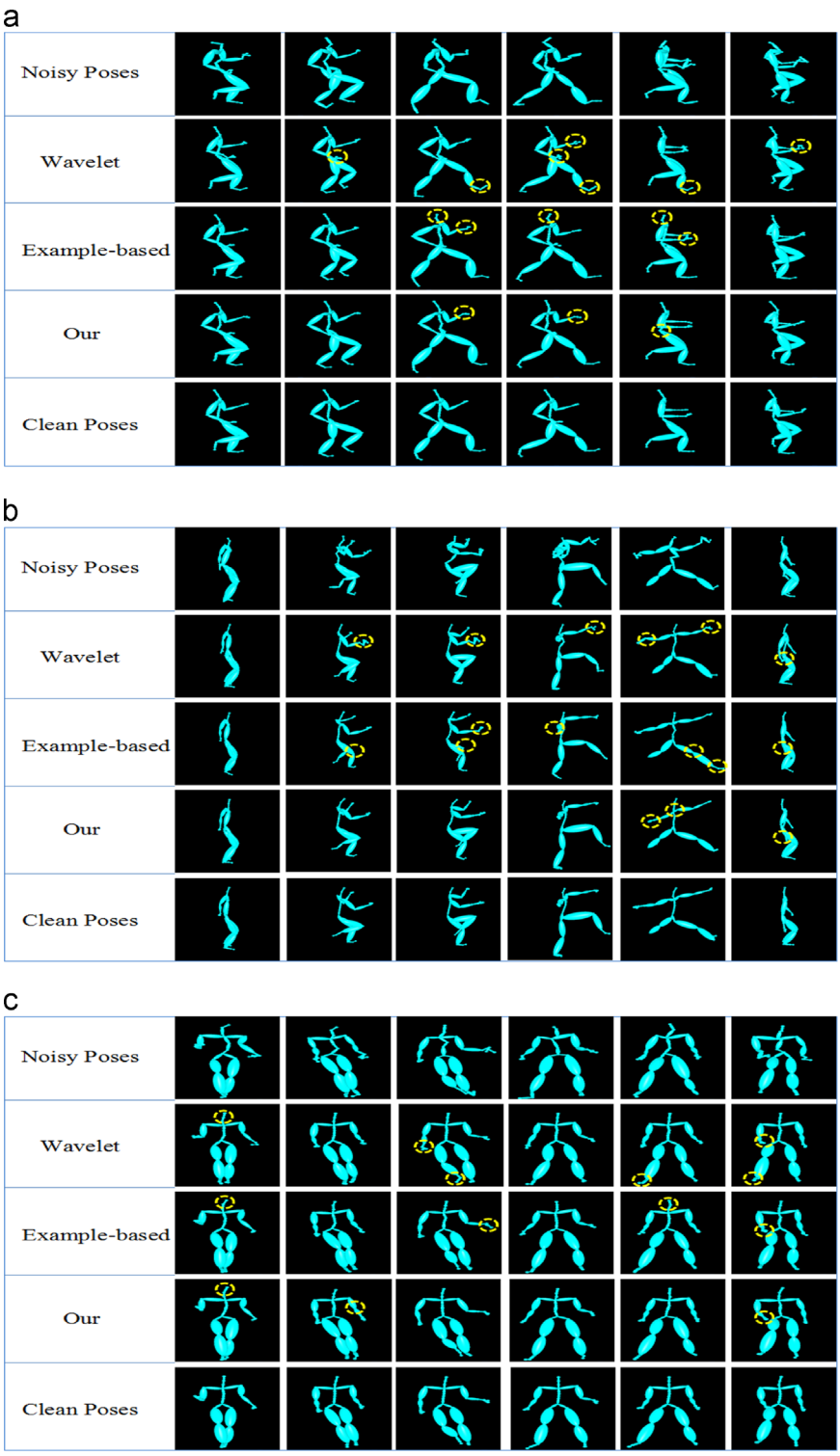

Fig. 8. Comparison results of some denoised key poses via the top-three algorithms (i.e., wavelet-based method, the example-based method and our method) on three human motion sequences. In each figure, the markers with a large deviation from its original location (>8 cm) are marked with yellow circles. (a) Comparison results of the denoised key poses via different algorithms on a run motion sequence, (b) comparison results of the denoised key poses via different algorithms on a dance motion sequence and (c) comparison results of the denoised key poses via different algorithms on a basketball motion sequence. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

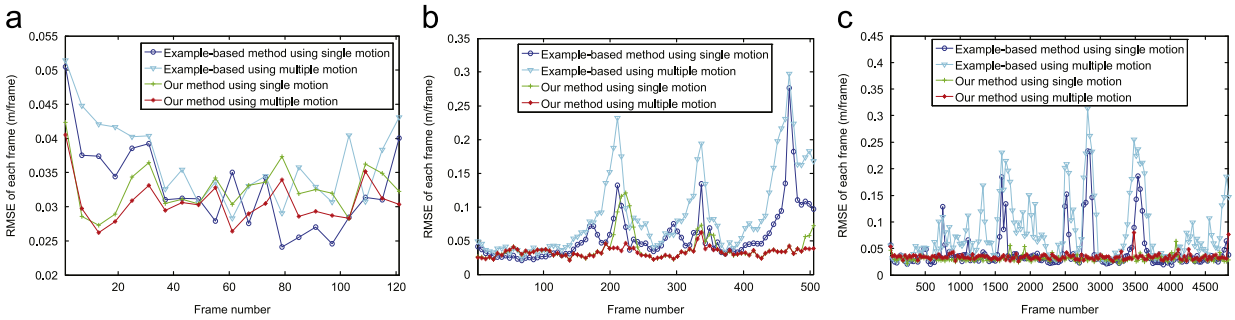

Fig. 9. Performance comparison of the two data-driven based denoising methods (i.e., our method and the example-based method) using the single motion or multiple motion data as the training data. (a) Run, (b) dance and (c) basketball.

Table 2
Running time and mean RMSE for our method using poselets representation and pose representation in different motion sequences. Here the units of time and mean RMSE are second and meter per frame separately.

Representation	Run		Dance		Basketball	
	Time	RMSE	Time	RMSE	Time	RMSE
Poselet	83.9	0.014	294.9	0.016	374.6	0.015
Pose	303.2	0.043	878.9	0.022	979.5	0.023

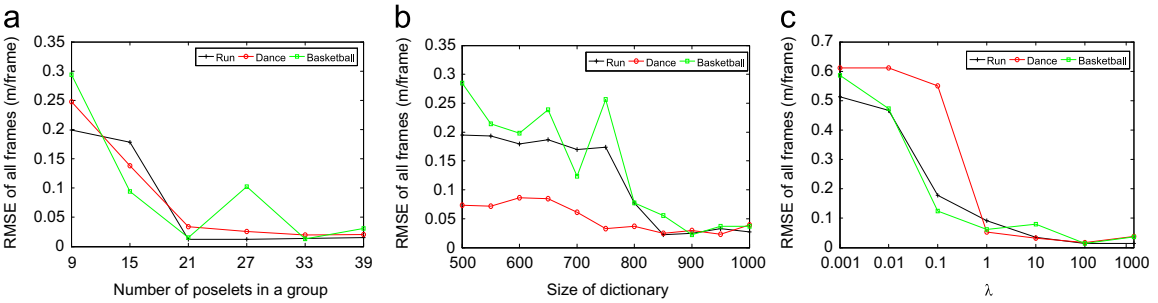

Fig. 10. Performance variance of our proposed method with respect to the three parameters (a) M, (b) K and (c) λ.

a general ℓ_1-minimization framework. For the two mostly common noises, i.e., Gaussian noise and outliers, two slightly different objective functions were derived from the same framework to exploit the statistic property of noise in motion data. We compared our method with other four methods. Experimental results demonstrated that the proposed method outperforms its competitors. Since our method does not need to specially choose the training data, it can be more easily applied to real-world applications. However, in order to train the motion dictionaries, our method needs some precaptured clean data as the training data, although it will be better that if we could robustly learn them directly from the unclean motion data. Therefore, we will investigate this issue and develop a robust dictionary learning method in the near future.

Acknowledgments

This research is supported by the National High Technology Research and Development Program (2012AA011502), the National Key Technology R&D Program (2013BAH59F00), the Zhejiang Provincial Natural Science Foundation of China (LY13F020001), the Fundamental Research Funds for the Central Universities (2014FZA5013), Zhejiang Province Public Technology Applied Research Projects (No. 2014C33090), and partially supported by the grant of the "Sino-UK Higher Education Research Partnership for Ph.D. Students" Project funded by the Department of Business, Innovation and Skills of the British Government and Ministry of Education of PR China.

References

[1] T.B. Moeslund, E. Granum, A survey of computer vision-based human motion capture, Comput. Vis. Image Underst. 81 (3) (2001) 231–268.
[2] T.B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, Comput. Vis. Image Underst. 104 (2) (2006) 90–126.
[3] R. Poppe, Vision-based human motion analysis: an overview, Comput. Vis. Image Underst. 108 (1–2) (2007) 4–18.

[4] L. Herda, P. Fua, R. Plaenkers, R. Boulic, D. Thalmann, Skeleton-based motion capture for robust reconstruction of human motion, in: Computer Animation, IEEE, Philadelphia, PA, USA, 2000, pp. 77–86.

[5] C.-C. Hsieh, Motion smoothing using wavelets, J. Intell. Robot. Syst. 35 (2) (2002) 157–169.

[6] L. Jehee, S.Y. Shin, General construction of time-domain filters for orientation data, IEEE Trans. Vis. Comput. Graph. 8 (2) (2002) 119–128.

[7] A. Van Rhijn, R. van Liere, J.D. Mulder, An analysis of orientation prediction and filtering methods for VR/AR, in: Proceedings of the 2005 IEEE Conference on Virtual Reality, IEEE, Bonn, Germany, 2005, pp. 67–74.

[8] H. Lou, J. Chai, Example-based human motion denoising, IEEE Trans. Vis. Comput. Graph. 16 (5) (2010) 870–879. ISSN 1077-2626.

[9] X. Wei, P. Zhang, J. Chai, Accurate realtime full-body motion capture using a single depth camera, ACM Trans. Graph. 31 (6) (2012). 188:1–12.

[10] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, R. Moore, Real-time human pose recognition in parts from single depth images, Commun. ACM 56 (1) (2013) 116–124.

[11] J. Lee, S.Y. Shin, Motion fairing, in: Computer Animation, IEEE, Geneva, Switzerland, 1996, pp. 136–143.

[12] Y. Fangt, C. Hsieh, M. Kim, J. Chang, T. Woo, Real time motion fairing with unit quaternions, Comput.-Aided Des. 30 (3) (1998) 192–198.

[13] M.P. Wachowiak, G.S. Rash, P.M. Quesada, A.H. Desoky, Wavelet-based noise removal for biomechanical signals: a comparative study, IEEE Trans. Biomed. Eng. 47 (3) (2000) 360–368.

[14] A.M. Wink, J.B. Roerdink, Denoising functional MR images: a comparison of wavelet denoising and Gaussian smoothing, IEEE Trans. Med. Imag. 23 (3) (2004) 374–387.

[15] G. Chen, T. Bui, A. Krzyzak, Image denoising using neighbouring wavelet coefficients, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04), vol. 2, IEEE, Amsterdam, The Netherlands, 2004, pp. 917–920.

[16] L. Li, J. McCann, N. Pollard, C. Faloutsos, BoLeRO: a principled technique for including bone length constraints in motion capture occlusion filling, in: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA), Eurographics Association, 2010, pp. 125–135.

[17] L. Li, J. McCann, N. Pollard, C. Faloutsos, Dynammo: mining and summarization of coevolving sequences with missing values, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD), ACM, 2009, pp. 507–516.

[18] Y. Feng, J. Xiao, Y. Zhuang, X. Yang, J.J. Zhang, R. Song, Exploiting temporal stability and low-rank structure for motion capture data refinement, Inf. Sci. 277 (1) (2014) 777–793.

[19] I. Akhter, T. Simon, S. Khan, I. Matthews, Y. Sheikh, Bilinear spatiotemporal basis models, ACM Trans. Graph. 31 (2) (2012). 17:1–12.

[20] A. Bruderlin, L. Williams, Motion signal processing, in: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, ACM, New York, NY, USA, 1995, pp. 97–104.

[21] K. Yamane, Y. Nakamura, Dynamics filter-concept and implementation of online motion generator for human figures, IEEE Trans. Robot. Autom. 19 (3) (2003) 421–432.

[22] C.-C. Hsieh, P.-L. Kuo, An impulsive noise reduction agent for rigid body motion data using B-spline wavelets, Expert Syst. Appl. 34 (3) (2008) 1733–1741.

[23] T. Tangkuampien, D. Suter, Human motion de-noising via greedy kernel principal component analysis filtering, in: The 18th International Conference on Pattern Recognition (ICPR), vol. 3, IEEE, Hong Kong, China, 2006, pp. 457–460.

[24] N.N. Schraudolph, S. Gunter, S. Vishwanathan, Fast iterative kernel PCA, Adv. Neural Inf. Process. Syst. 19 (2007) 1225–1232.

[25] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Netw. 13 (4) (2000) 411–430.

[26] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.

[27] D. Tao, X. Tang, X. Li, X. Wu, Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 28 (7) (2006) 1088–1099.

[28] D. Tao, X. Li, X. Wu, S.J. Maybank, General tensor discriminant analysis and Gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1700–1715.

[29] D. Tao, X. Li, X. Wu, S.J. Maybank, Geometric mean for subspace selection, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 260–274.

[30] W. Wang, M.A. Carreira-Perpinán, Manifold blurring mean shift algorithms for manifold denoising, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, San Francisco, CA, 2010, pp. 1759–1766.

[31] R. Gonsalves, J. Teizer, Human motion analysis using 3D range imaging technology, in: The 26th International Symposium on Automation and Robotics in Construction (ISARC), 2009, pp. 76–85.

[32] S. Tak, H.-S. Ko, A physically-based motion retargeting filter, ACM Trans. Graph. 24 (1) (2005) 98–117.

[33] H.J. Shin, J. Lee, S.Y. Shin, M. Gleicher, Computer puppetry: an importance-based approach, ACM Trans. Graph. 20 (2) (2001) 67–94.

[34] J.M. Wang, D.J. Fleet, A. Hertzmann, Gaussian process dynamical models for human motion, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2) (2008) 283–298.

[35] R. Lai, P. Yuen, K. Lee, Motion capture data completion and denoising by singular value thresholding, in: Eurographics, The Eurographics Association, Llandudno, UK, 2011, pp. 45–48.

[36] J. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. 20 (4) (2010) 1956–1982.

[37] A. Ganesh, S.S. Sasttry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Learn. 31 (2) (2009) 210–227.

[38] J. Xiao, Y. Feng, W. Hu, Predicting missing markers in human motion capture using l1-sparse representation, Comput. Anim. Virtual Worlds 22 (2–3) (2011) 221–228.

[39] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process. 15 (12) (2006) 3736–3745.

[40] R. Ron, Z. Michael, M. Elad, Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit, Technion-Computer Science Department, Technical Report, CS-2008-08.

[41] K. Skretting, K. Engan, Recursive least squares dictionary learning algorithm, IEEE Trans. Signal Process. 58 (4) (2010) 2121–2130.

[42] R. Rubinstein, T. Faktor, M. Elad, K-SVD dictionary learning for the analysis sparse model, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Kyoto, Japan, 2012, pp. 5405–5408.

[43] C. Hong, Z. Liu, L. Yang, X. Chen, Sparse representation and learning in visual recognition: theory and applications, Signal Process. 93 (6) (2013) 1408–1425.

[44] H. Lee, A. Battle, R. Raina, A.Y. Ng, Efficient sparse coding algorithms, in: Advances in Neural Information Processing Systems, 2006, pp. 801–808.

[45] J. Yu, D. Tao, M. Wang, Adaptive hypergraph learning and its application in image classification, IEEE Trans. Image Process. 21 (7) (2012) 3262–3272.

[46] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, IEEE Trans. Image Process. 23 (5) (2014) 2019–2032.

[47] J. Yu, R. Hong, M. Wang, J. You, Image clustering based on sparse patch alignment framework, Pattern Recognit 47 (2014) 3512–3519.

[48] I. Tosic, P. Frossard, Dictionary learning, IEEE Signal Process. Mag. 28 (2) (2011) 27–38.

[49] D. Bartuschat, A. Borsdorf, H. Köstler, R. Rubinstein, M. Stürmer, A Parallel K-SVD Implementation for CT Image Denoising, Department of Computer Science, 2009, pp. 2—26.

[50] V. Sindhwani, A. Ghoting, Large-scale distributed non-negative sparse coding and sparse dictionary learning, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), ACM, New York, NY, USA, 2012, pp. 489–497.

[51] R. Rubinstein, T. Peleg, M. Elad, Analysis K-SVD: a dictionary-learning algorithm for the analysis sparse model, IEEE Trans. Signal Process. 61 (2013) 661–677.

[52] J. Baumann, K. Björn, A. Zinke, A. Weber, Data-driven completion of motion capture data, in: Workshop on VRIPHYS, Eurographics Association, Lyon, France, 2011, pp. 111–118.

[53] D.L. Donoho, Y. Tsaig, I. Drori, J.-L. Starck, Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit, IEEE Trans. Inf. Theory 58 (2) (2012) 1094–1121.

[54] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1) (1998) 33–61.

[55] I.F. Gorodnitsky, B.D. Rao, Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm, IEEE Trans. Signal Process. 45 (3) (1997) 600–616.
[56] G.H. Golub, C.F. Van Loan, An analysis of the total least squares problem, SIAM J. Numer. Anal. 17 (6) (1980) 883–893.
[57] M. Schmidt, Least Squares Optimization with L1-norm Regularization, Technical Report, Project Report, University of British Columbia, 2005.
[58] P.W. Holland, R.E. Welsch, Robust regression using iteratively reweighted least-squares, Commun. Stat. Theory Methods 6 (9) (1977) 813–827.
[59] J. Yang, Y. Zhang, Alternating direction algorithms for ℓ_1-problems in compressive sensing, SIAM J. Sci. Comput. 33 (1) (2011) 250–278.
[60] A.Y. Yang, Z. Zhou, A.G. Balasubramanian, S.S. Sastry, Y. Ma, Fast-minimization algorithms for robust face recognition, IEEE Trans. Image Process. 22 (8) (2013) 3234–3246.
[61] Carnegie Mellon University Graphics Lab Motion Capture Database ⟨http://mocap.cs.cmu.edu⟩, 2014.