Information Sciences 277 (2014) 777–793

Contents lists available at ScienceDirect

Information Sciences

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

Exploiting temporal stability and low-rank structure for motion capture data refinement

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

[a] School of Computer Science, Zhejiang University, Hangzhou 310027, PR China
[b] National Centre for Computer Animation, Bournemouth University, Poole, United Kingdom

ARTICLE INFO

Article history:
Received 7 May 2013
Received in revised form 17 February 2014
Accepted 5 March 2014
Available online 13 March 2014

Keywords:
Motion capture data
Data refinement
Matrix completion
Temporal stability

ABSTRACT

Inspired by the development of the matrix completion theories and algorithms, a low-rank based motion capture (mocap) data refinement method has been developed, which has achieved encouraging results. However, it does not guarantee a stable outcome if we only consider the low-rank property of the motion data. To solve this problem, we propose to exploit the temporal stability of human motion and convert the mocap data refinement problem into a robust matrix completion problem, where both the low-rank structure and temporal stability properties of the mocap data as well as the noise effect are considered. An efficient optimization method derived from the augmented Lagrange multiplier algorithm is presented to solve the proposed model. Besides, a trust data detection method is also introduced to improve the degree of automation for processing the entire set of the data and boost the performance. Extensive experiments and comparisons with other methods demonstrate the effectiveness of our approaches on both predicting missing data and de-noising.

1. Introduction

In recent years, with the rapid development of motion capture (mocap) techniques and systems, motion capture data have been widely used in computer games, film production and sport sciences [38,39,42,52]. The great success of animated and animation enhanced feature films *Avatar* provides a compelling evidence for the values of mocap techniques. However, even with the most expensive commercial mocap systems, there are still instances where noise and missing data are inevitable [1,3,18,21]. Fig. 1 presents some real examples captured by ourselves using an optical mocap system. It can be seen that the acquired raw mocap data contains imperfections which should be refined before used for animation production. Thus, an important branch of motion capture research focuses on handling two highly correlated and frequently co-occurred sub-problems: one is to predict the missing values in mocap data and the other is to remove both the noise and outliers. These two sub-problems are collectively referred to as *mocap data refinement problem*. Meanwhile, the prevalence of some novel yet cheap mocap sensors (e.g., Microsoft Kinect), which are able to capture human motion in real-time but whose outputs are not very accurate especially when significant occlusions occur, makes the mocap data refinement problem much more pertinent and important [13,45,48,54,60].

[*] Corresponding author. Tel.: +86 13867424906.
E-mail addresses: fyf200502@hotmail.com (Y. Feng), junx@zjuem.zju.edu.cn (J. Xiao), yzhuang@zju.edu.cn (Y. Zhuang), xyang@bournemouth.ac.uk (X. Yang), jzhang@bournemouth.ac.uk (J.J. Zhang), srypl@zju.edu.cn (R. Song).

http://dx.doi.org/10.1016/j.ins.2014.03.013

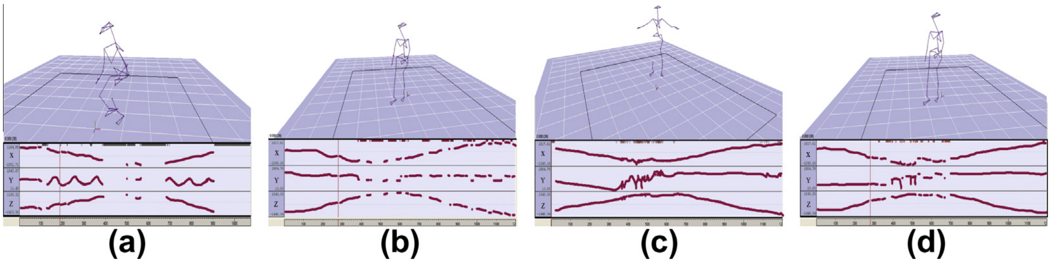


Fig. 1. Four real examples of mocap data with missing and noisy values from our own captured motion sequences. In each subgraph, one joint of human skeleton is selected and its trajectory's *x*-, *y*-, *z*-axis curves are plotted below.

Although many researchers have studied this problem and numerous techniques have been proposed to deal with this problem, often the performance and application of these methods are hindered by their inherent drawbacks. For example, the interpolation methods used in some commercial mocap systems such as EVaRT and Vicon are only suitable to deal with the short-time (<0.5 s) data missing problem [1]. The data-driven methods often suffer from the out-of-sample problem [4,5,37]. It is not exaggerated to say that the refinement problem is far from being solved and has attracted a great deal of attention in recent years [1,2,4,5,12,19,23,37,47,58].

To handle this problem, a new optimization method is presented to process the input motion clip directly without the support of any motion database. We convert the mocap data refinement problem into a robust matrix completion problem with a smooth constraint. The proposed Temporal Stable and noise robust Matrix Completion (TSMC) model takes into account both the low-rank structure and temporal stability properties of motion data as well as the effect of noise. Additionally, we present an efficient trust data detection method to automate the data processing and boost the performance. The overview of our proposed approach is shown in Fig. 2.

Briefly speaking, our research makes three main contributions:

1. We can successfully convert the traditional mocap data refinement problem into a robust matrix completion problem with the smooth constraint and propose a new model to exploit both the low-rank structure and the temporal stability properties of mocap data at the same time.
2. We present a fast optimization method to solve the derived mathematic model with its convergence guaranteed theoretically.
3. An efficient trust data detection method is designed to increase the degree of automation for data processing and boost the performance of our model.

In the remainder of this paper, we will first review some closely related work in Section 2. Then the detail of our proposed approach is given in Section 3 with extensive experiments following in Section 4. Finally, the paper is concluded in Section 5.

2. Related work

When motion data is acquired using a mocap system, data refinement is an indispensable subsequent post-process step. More precisely, in this step two sub-problems must be solved: one is to predict the missing values in mocap data and the other is to remove the noise and outliers. These two seemingly separate issues often co-occur in practice, although they have been discussed largely separately in the literature. For the purpose of our discussion, we briefly review some of the closely related work on these two sub-problems.

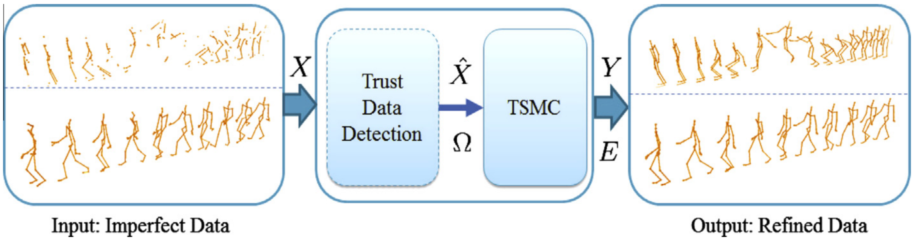


Fig. 2. Overview of our proposed approach for handling the motion capture data refinement problem. The approach contains two main parts: (1) detect and find out the trust data entries using an efficient trust data detection method and then set $\hat{X}$ and Ω according to the detection result and (2) refine the imperfect data using the proposed TSMC algorithm.

Due to reasons such as markers falling off or occlusion, the loss of marker information is a frequent challenge in human motion capture. How to predict missing values in mocap data has been studied by many researchers [1,4,17,23,30,31,57,58] in the past twenty years. The interpolation methods are widely used owing to the advantages that they are very fast and easy to implement. However, they are off-line methods and only efficient for handling missing values for a very short period of time, typically less than 0.5 s [1]. To overcome these drawbacks, Herda et al. [17] introduced a real-time method using an anatomical human model to predict the position of the markers. It is unfortunately very difficult and time consuming to set-up such a model. Aristidou et al. [1] used Kalman Filters to estimate missing markers in real-time without the support of any human model. However, the filtered human motion exhibits visible short latency. Moreover, Kalman Filters may fail when markers are missing or the motion data is corrupted by noise for an extended time period [12]. To discover hidden variables and learn their dynamics, Li et al. [30] built a probabilistic model to estimate the expectation of missing values conditioned on the observed parts. Their next work [31] imposed bone-length constraints in a linear dynamical system (LDS) to boost the performance. The method [31], however, has to rely on the existence of other markers on the same segment to make inter-marker distance measurement possible [4].

Recently, data-driven methods have attracted a lot of attention. Xiao et al. [58] adopted sparse representation to predict the missing values and Baumann et al. [4] fixed missing data via searching poses with a similar marker set from a prior-database and then optimizing an energy minimization function to synthesize the positional data of the missing markers. The shortage of data-driven methods is the out-of-sample problem and they require an additional data preprocessing step which will increase the human labor cost.

The out-of-sample problem will be overcome, if one can refine mocap data without the support of a database. Lai et al. [23] noticed that the low-rank property of mocap data has not been explicitly exploited in the previous work, and they proposed to handle the mocap data refinement problem based on low-rank matrix completion theory and algorithm. The key point of their method is that it does not need any training data. Inspired by [23], our model also takes the low-rank structure property into account. Besides, we include two other properties into our design: the temporal stability and the noise effect. Compared with [23], our model does not only take the low-rank structure property into account but also the temporal stability property of motion data and noise effect. Our proposed model can handle both sub-problems of mocap data refinement at the same time, while in [23] they used two separate models to achieve the same goal. Another significant difference between our model and its counterpart in [23] is that we do not need to guess the standard deviation of the noise which is used for solving the de-noising model in [23]. More importantly, the optimization method for solving our model is much faster and robust than SVT [23], which has been proven both in theory and in experiment [33]. In Section 4, we also have observed that our method is not only faster than the work [23] but also outperforms it in the experiments on both synthetic and real data. Meanwhile, we also notice that two low-rank matrix based methods [19,47] have been proposed almost at the same time as ours following the work [23]. However, both papers [19,47] aim to well use the low-rank property of the trajectory matrix of human motion data by reorganizing mocap data into sequences of trajectory segments. Moreover, they can be used to handle just one sub-problem of mocap data refinement (i.e., predicting missing values), while our method can simultaneously solve both sub-problems.

There are also a lot of research focusing on removing the noise and outliers from mocap data, which is called mocap data de-noising [20,23,24,26,27,37]. Lee and Shin [26] formulated rotation smoothing as a nonlinear optimization problem and iteratively minimized the energy function to smooth the motion. In their later work [27], they proposed a linear time-invariant filtering framework for filtering the orientation data by transforming the orientation data into their analogues in a vector space; applying a time-domain filter on them; and transforming the results back to the orientation space. Lou and her colleagues [37] learned a series of filter bases from prerecorded mocap data and then used them in a robust nonlinear optimization framework to perform mocap data denoising. In [23], Lai et al. simply modified the objective function of SVT [7] by imposing an inequality equation so that it is able to handle the mocap data de-noising problem. However, in their work, the user has to guess the standard deviation of noise, which is difficult in practice. Moreover, only taking the low-rank property into account, the recovered motion is not so stable that some poses will shake. Some corrupted markers cannot be recovered.

3. Our approach

3.1. Notations

To better present the details of our approach, we provide some important notations used in the rest of this paper. Capital letters, e.g., X, represent matrices or sets, $X_{i,:}$ is the i-th row of X and $X_{:,j}$ is the j-th column of X. Lower case letters, e.g., x, represent vectors or scalar values. Superscript (i), e.g., $X^{(i)}$ and $x^{(i)}$, represents datum in the i-th iteration. Throughout this paper, I_c denotes the $c \times c$ identity matrix; $\Omega \in \{0,1\}^{m\times n}$ is a binary matrix and $\overline{\Omega}$ is the corresponding complement matrix, i.e., $\overline{\Omega}_{ij} = 1 - \Omega_{ij}$. For any matrix $X \in \mathcal{R}^{p\times q}$, let $\|X\|_0$ be the l_0-norm, $\|X\|_1 = \sum_{i,j}|X_{ij}|$ be the l_1-norm, $\|F\|_2 = \|X\|_F = \left(\sum_{i,j} X_{ij}^2\right)^{\frac{1}{2}}$ be the l_2-norm and the Frobenius norm, $\|X\|_\infty = \max_{1\leqslant i\leqslant p}\sum_{j=1}^{q}|X_{ij}|$ is the ∞-norm, and $\|X\|_* = \sum_{i=1}^{r}\sigma_i(X)$ be the nuclear norm, where $r = \min\{p,q\}$ and $\sigma_i(X)$ be the i-th largest singular value of X. X^T stands for the transpose of X. Additionally, we denote $X \circ Y$ be the Hadamard product of X and Y, i.e., $X \circ Y = [X_{ij}Y_{ij}]$, and $\langle X, Y\rangle = tr(X^TY)$, wherein $tr(\cdot)$ is the matrix trace operation.

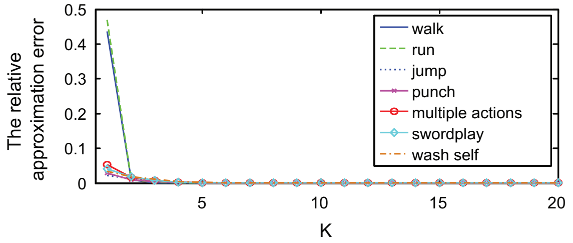


Fig. 3. The relative approximation error of various motion capture data.

3.2. Objective function

Based on the above defined notations, we denote a motion sequence which consists of n poses (frames) as $\mathcal{X} = \{f_1, f_2, \ldots, f_n\}$, where $f_i \in \mathcal{R}^d$ is the i-th pose. In this work we use the 3D coordinates (i.e., (x, y, z)) of markers to represent a pose feature, i.e., $f_i = [x_1, y_1, z_1, \ldots, x_k, y_k, z_k]^T$, where k is the number of markers (i.e., $d = 3 \times k$).

Intuitively, the nearby poses in $\mathcal{X}$ are often similar to each other and the motion sequence is of high temporal correlation. In other words, if we represent this sequence in a motion matrix form $X = [f_1, f_2, \ldots, f_n] \in \mathcal{R}^{d \times n}$, it is naturally to guess that X should have a low-rank structure [23]. To verify this hypothesis, we first collect 7 motion sequences with diverse activities from the CMU mocap dataset.[1] Then, we centralize X and apply singular value decomposition (SVD) on X^TX to examine whether the mocap data has a good low-rank approximation. The metric we used here is the relative approximation error of the top K biggest singular values, i.e., $1 - \left(\sum_i^K \sigma_i^2\right) / \left(\sum_i^r \sigma_i^2\right)$, where σ_i is the i-th largest singular value and $\left(\sum_i^r \sigma_i^2\right)$ gives the total variance of the matrix X^TX [43,50]. As we can see from Fig. 3, the top five biggest singular values can well approximate the whole variance in all of the diversity activities. This result suggests that mocap data exhibits a low-rank structure.

Additionally, natural human motion satisfies the temporal stability constraint, i.e., the motion trajectories of markers are smooth most of the time [9,44,49]. As shown in Fig. 4, it is smooth and stable in the direction of frame index (i.e., the temporal direction which is the x-axis in Fig. 4). We believe there are two reasons leading to this situation. One is because that the velocities of each markers are stable in a short period of time. The second reason is that the capture speed of the current commercial mocap systems is very high (more than 60–120 fps[2]) and therefore smooth and stable human motions can be recorded accurately.

To further confirm our view, we calculate the velocity (i.e., $V_{i,j} = X_{i,j+1} - X_{i,j}$) and acceleration (i.e., $A_{i,j} = \frac{V_{i,j+1} - V_{i,j}}{\Delta t}$ and $\Delta t = 1$ in our case) for each feature dimension and plot the results in Fig. 5(a) and (b). We find that although the velocities of different pose feature dimensions are diverse, their accelerations are very small (<±1 cm/frame^2)[3] mostly in a short period of time. Ideally, if the articulation joints move with some constant velocities $\mathbf{v}$, the coordinate values of all these markers at time t can be represented as $f_t = f_0 + t \cdot \mathbf{v}$, where f_0 is the initial coordinate vector. Here we denote $\mathbf{1} = [1, \ldots, 1]^T$ and $\mathbf{t} = [1, 2, \ldots, n]^T$. Then, the generated motion matrix $X = [f_1, \ldots, f_n] = f_0 \cdot \mathbf{1}^T + \mathbf{v} \cdot \mathbf{t}^T$ is both low-rank (the rank is 2) and temporal stable, even in the case that each markers with different velocities.

Meanwhile, Candès et al. [8] proved that it is possible to recover most low-rank matrices from what appears to be an incomplete set of entries. If an imperfect motion sequence is refined by matrix completion methods, the result also should be both low-rank and temporally stable. In order to enhance the robustness of our model, we only assume that the noise is sparse in the observed part. It is a weak but reasonable assumption, because the observable data from the current mocap systems usually contain a little amount of noise and outliers. It means only a small percentage of the available data is corrupted. Here we propose a robust low-rank matrix completion model to well exploit the low-rank structure and temporal stability properties of mocap data as follows:

$$\min_{Y,E} \quad rank(Y) + \alpha \|\Omega \circ E\|_0 + \frac{\beta}{2} \Theta(Y), \tag{1}$$

$$\text{s.t.} \quad Y + E = X, \tag{2}$$

where $\Omega \in \{0,1\}^{d \times n}$ is a mask matrix (i.e., $\Omega_{i,j} = 1$ for the observable $X_{i,j}$ and $\Omega_{i,j} = 0$ for the missing entry $X_{i,j}$). X is the imperfect motion data and Y is the corresponding complete and clean motion data. Here $\Theta(Y)$ is a smooth penalty function and E is the sparse noise or outliers. When some markers are missing and we denote these missing values as zeros, E contains the opposite numbers[4] of the corresponding missing values.

[1] http://mocap.cs.cmu.edu/.

[2] Now, the capture speed of Eagle-4 high speed video camera can achieve up to 500 fps at 1280 × 1024 full resolution.

[3] Note that it is an approximate acceleration used here. Indeed, the real acceleration is $\pm 1\ \text{cm}/(\frac{1}{60}\ \text{s}^2) = \pm 0.6\ \text{m/s}^2$.

[4] Opposite number of any number x is a number which if added to x results in 0. The opposite number for x is usually denoted as $-x$.

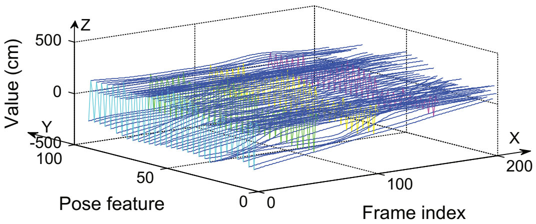
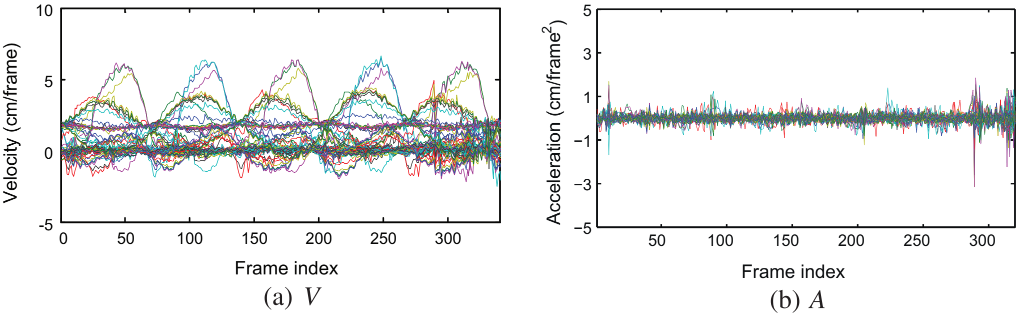


Fig. 4. Trajectories from a walk motion sequence. Here the x-axis represents the frame index, which is the column number of X, and the y-axis represents the index of the pose feature, which is the row number of X, and the z-axis represents the values in each corresponding position of X. The blue curves represent the trajectories of each pose feature (i.e., $X_{i,:}, \forall i$) and the other color curves link different features at the frames of $1, 50, 100, 150$ and 200 (i.e., $X_{:,j}, j = 1, 50, 100, 150, 200$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) V

(b) A

Fig. 5. The velocity and acceleration of each feature dimension in a walk motion sequence.

Because Y is smooth in the temporal direction as shown in Fig. 4, we enforce C^2 continuity on each feature dimension of Y, i.e., $\forall r, Y_{r,:}$ is C^2 continuity. This requirement leads to $\Theta(Y) = \|YO^T\|_F^2$ [10,14,28], and O is tridiagonal square matrix defined by

$$O_{i,i-1} = \frac{2}{h_{i-1}(h_{i-1} + h_i)}, \tag{3}$$

$$O_{i,i} = \frac{-2}{h_{i-1}h_i}, \tag{4}$$

$$O_{i-1,i} = \frac{2}{h_i(h_{i-1} + h_i)}, \tag{5}$$

for $\forall i,\ 2 \leqslant i \leqslant n-1$, where n is the number of elements in $Y_{r,:}$ and h_i represents the step between $Y_{r,i}$ and $Y_{r,i+1}$. Assuming repeating border elements, that is $Y_{r,0} = Y_{r,1}$ and $Y_{r,n+1} = Y_{r,n}$, gives $O_{1,1} = -O_{1,2} = -1/h_1^2$ and $O_{n,n-1} = -O_{n,n} = -1/h_{n-1}^2$. In our special case, the data can be treated as equally spaced which means $\forall i,\ h_i = 1$, so we get:

$$O = \begin{bmatrix} -1 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -1 \end{bmatrix}. \tag{6}$$

Note that O is a symmetrical matrix, Eq. (1) can be equivalently expressed as follows.

$$\min_{Y,E} \quad rank(Y) + \alpha\|\Omega \circ E\|_0 + \frac{\beta}{2}\|YO\|_F^2. \tag{7}$$

Since Eq. (7) is a l_0-norm regularized rank-minimization problem, which is NP-hard, in most matrix completion problems the l_1-norm and the nuclear norm are often used as surrogates of matrix rank and the l_0-norm respectively [7,11,53,56]. So, we relax Eq. (7) and get the model:

$$\min_{Y,E} \quad \|Y\|_* + \alpha\|\Omega \circ E\|_1 + \frac{\beta}{2}\|YO\|_F^2, \tag{8}$$

$$\text{s.t.} \quad Y + E = X, \tag{9}$$

where α and β are two regularization parameters. In the above model (i.e., Eq. (8)), the first nuclear norm term restricts the refined motion Y to be low-rank, the second term takes the effect of sparse noise into account and the last one incorporates the temporal stability constraint into our model. We call it the Temporal Stable and noise robust Matrix Completion (TSMC) model.

3.3. Trust data detection

In our proposed TSMC model, apart from the parameters α and β, we also need to set two matrices: Ω and O. As O depends on the dimension of X, it is easy to set. If we know X only contains missing values and some relatively small noise, we can simply set $\Omega_{i,j} = 1$ if $X_{i,j}$ is observable, otherwise $\Omega_{i,j} = 0$. But X is frequently contaminated by some noise and outliers in practice. Thus it is unwise even incorrect to simply set $\Omega_{i,j}$ based on whether $X_{i,j}$ is observable or not. A more sophisticated choice is to detect and find out the trust data entries first, and then set Ω according to these trust data as shown in Fig. 2. In the following part, we propose an efficient method to find out the trust entries in X based on the smoothness property of human motion data in each pose feature dimension, as mentioned earlier.

Let's consider the following model for the one-dimensional noisy signal x:

$$x = \hat{x} + \epsilon, \tag{10}$$

where ϵ represents some noise and $\hat{x}$ is a smooth function. To smooth the data, a common choice is to enforce C^2 continuity on $\hat{x}$, in which case $\hat{x}$ is called a (cubic) spline. And inspired by the success of the penalized least squares regression model [14], we seek to minimize the following objective function

$$\mathcal{L}(\hat{x}) = \|G^{1/2}(\hat{x} - x)\|^2 + \mu\mathcal{R}(\hat{x}), \tag{11}$$

where $\mathcal{R}(\hat{x})$ is a regularization term, μ is a scalar which determines the balance between the first fitting term and the second smoothing term, and G is a diagonal matrix such that $G_{i,i}$ represents a weight assigned to the observed ith entry of x. The main drawback of the penalized least squares regression models is their sensitivity to the outliers. We should choose robust weighting functions to minimize or cancel the side effects of the outliers. Several weighting functions are available to achieve this goal, such as the bisquare weight function [14,16] and the Welsch robust function [37] and so on. In order to mitigate the effect of noise and outliers, we define G as follows:

$$G_{i,i} = \begin{cases} \exp\left(-\frac{r_i^2}{p^2}\right) & \text{if } x_i \text{ is observable,} \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where $r_i = x_i - \hat{x}_i$ is the residual, p is a scalar and we experimentally fix it to 5.

From Section 3.2, we know that the C^2 continuity requirement has led to the definition of $\mathcal{R}(\hat{x}) = \|O\hat{x}\|^2$. Thus Eq. (11) can be reformulated as

$$\mathcal{L}(\hat{x}) = \|G^{1/2}(\hat{x} - x)\|^2 + \mu\|O\hat{x}\|^2. \tag{13}$$

Solving Eq. (13), the minimization of Eq. (11) is given by

$$\begin{aligned} \hat{x}^{(k+1)} &= (I + \mu O^T O)^{-1}(G^{(k)}(x - \hat{x}^{(k)}) + \hat{x}^{(k)}) && (14) \\ &= H(G^{(k)}(x - \hat{x}^{(k)}) + \hat{x}^{(k)}), && (15) \end{aligned}$$

where $H = (I + \mu O^T O)^{-1}$. Because $(I + \mu O^T O)$ is a sparse symmetric matrix, Eq. (14) can be efficiently solved [15,32].

As indicated by Eq. (14), the output $\hat{x}$ relies on the smoothing regularization parameter μ. In order to set it correctly and automatically, we resort to the method of generalized cross-validation (GCV) [10,28]. The GCV method chooses μ that minimizes the GCV score in the presence of weighted data as follows:

$$\mu = \arg\min_{\mu}(GCV(\mu)), \tag{16}$$

$$GCV(\mu) \equiv \frac{\|G^{1/2}(\hat{x} - x)\|^2/(n - n_s)}{(1 - tr(H)/n)^2}, \tag{17}$$

where n is the length of x and n_s is the number of missing entries. It is to be noticed that $tr(H)$ is equal to $\sum_{i=1}^{n}\frac{1}{1+\mu\sigma_i^2}$, where σ_i^2 are the eigenvalues of $O^T O$. Therefore, the GCV score is

$$GCV(\mu) = \frac{n^2\sum_{i=1}^{n}\sqrt{G_{i,i}}(\hat{x}_i - x_i)^2}{(n - n_s)\left(n - \sum_{i=1}^{n}(1 + \mu\sigma_i^2)^{-1}\right)}. \tag{18}$$

Till now, μ and $\hat{x}$ can be estimated based on Eqs. (16) and (14) iteratively. Suppose we get the optimal $\hat{x}_i \in \mathcal{R}^n, i = 1, \ldots, d$ for each feature dimension, the smoothed X can be represented as $\widehat{X} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_d]^T$. Then, the mask matrix Ω can be set as

$$\Omega_{i,j} = \begin{cases} 1 & \text{if } X_{ij} \text{ is observable and } |X_{ij} - \widehat{X}_{ij}| \leqslant \theta, \\ 0 & \text{otherwise}, \end{cases} \tag{19}$$

where θ is a threshold value which can be estimated from Fig. 5(a) and we set it to 6 cm in our experiments. Also, if a large percentage of X is badly corrupted by noise, we can filter X using the smoothed $\widehat{X}$ as follows:

$$\widetilde{X}_{i,j} = \begin{cases} X_{ij} & \text{if } |X_{ij} - \widehat{X}_{ij}| \leqslant \theta, \\ \widehat{X}_{ij} & \text{otherwise}. \end{cases} \tag{20}$$

In summary, the proposed trust data detection method is listed in Algorithm 1.

Algorithm 1. Trust Data Detection Method

```
Input:   $x, \theta$;
Output   $\hat{x}, \Omega$;
1: Initialize:
   $i = 0, \hat{x}^{(0)} = \mathbf{0}$;
2: Repeat:
3:      Calculate $G^{(k)}$ according to Eq. (12);
4:      Find $\mu$ according to Eq. (16);
5:      Update $\hat{x}^{(k+1)}$ according to Eq. (14)
6:   untile convergence or reach the maximum iteration times.
7: Set $\Omega$ according to Eq. (19)
```

3.4. Optimization method

To solve the proposed TSMC model, we first introduce a slack matrix $M = Y$ to decouple the terms containing Y in Eq. (8), and our model equivalent to:

$$\min_{Y,E,M} \quad \|Y\|_* + \alpha\|\Omega \circ E\|_1 + \frac{\beta}{2}\|MO\|_F^2, \tag{21}$$

$$\text{s.t.} \quad Y + E = X, \tag{22}$$

$$Y = M. \tag{23}$$

Indeed, the above low-rank matrix completion problem Eq. (21) can be efficiently solved via many algorithms such as the accelerated proximal gradient (APG) algorithm [34,55], the SVT algorithm [7] and the augmented Lagrange multiplier (ALM) algorithm [33]. In [33], the authors have compared all of these algorithms and demonstrated that the ALM algorithm is much faster than the other methods and its precision is also higher [33]. More importantly, the ALM algorithm has a pleasing Q-linear convergence speed. Therefore, we adopt the ALM algorithm to solve the above objective functions. And, the corresponding augmented Lagrange function is

$$\mathcal{J}(Y, E, M, Y_1, Y_2) = \|Y\|_* + \alpha\|\Omega \circ E\|_1 + \frac{\beta}{2}\|MO\|_F^2 + \langle Y_1, X - Y - E\rangle + \frac{\lambda_1}{2}\|X - Y - E\|_F^2 + \langle Y_2, M - Y\rangle + \frac{\lambda_2}{2}\|Y - M\|_F^2. \tag{24}$$

The ALM algorithm solves Eq. (24) by alternately minimizing $\mathcal{J}$ w.r.t. Y, E, M and maximizing it w.r.t. Y_1 and Y_2. Here we denote the singular value shrinkage operator $\mathcal{S}_\tau$ [7,33,53] as

$$[\mathcal{D}_\tau(A)]_{ij} = sgn(A_{ij})(|A_{ij}| - \tau)_+, \tag{25}$$

$$\mathcal{S}_\tau(B) = U_B\mathcal{D}_\tau\left(\sum\nolimits_B\right)V_B^T, \tag{26}$$

where $B = U_B\sum_B V_B^T$ is the SVD of B. Then the optimization method for solving Eq. (24) can be described in Algorithm 2. To increase the readability of this paper for the casual reader and save the space, the derivation details of Algorithm 2 can be found in Appendix A. For more details about the ALM algorithm, please refer to [33].

Algorithm 2. The Optimization Method for TSMC

```
**Input** $X, \Omega, O$, parameters $\alpha, \beta$ and $\gamma$;
**Output** $Y$ and $E$;
**Initialize**:
  $i = 0, Y_1^{(0)} = Y_2^{(0)} = X/max(\|X\|_2, \|X\|_\infty)$;
  $Y^{(0)} = \mathbf{0}, E^{(0)} = \mathbf{0}, M^{(0)} = \mathbf{0}$,
  $\lambda_1 > 0, \lambda_2 > 0$;
**Repeat:**
    $Z_Y \leftarrow \frac{Y_1^{(i)} + Y_2^{(i)} + \lambda_1^{(i)}(X - E^{(i)}) + \lambda_2^{(i)} M^{(i)}}{\lambda_1^{(i)} + \lambda_2^{(i)}}$
    $Y^{(i+1)} \leftarrow \mathcal{S}_{1/\left(\lambda_1^{(i)} + \lambda_2^{(i)}\right)}(Z_Y)$
    $Z_e \leftarrow \frac{1}{\lambda_1^{(i)}} Y_1^{(i)} + X - Y^{(i+1)}$;
    $E^{(i+1)} \leftarrow \Omega \circ \mathcal{D}_{\alpha/\lambda_1^{(i)}}(Z_e) + \overline{\Omega} \circ Z_e$
    $M^{(i+1)} \leftarrow (\lambda_2^{(i)} Y^{(i+1)} - Y_2^{(i)})(\beta O O^T + \lambda_2^{(i)} I)^{-1}$
    $Y_1^{(i+1)} \leftarrow Y_1^{(i)} + \lambda_1^{(i)}(X - Y^{(i+1)} - E^{(i+1)})$
    $Y_2^{(i+1)} \leftarrow Y_2^{(i)} + \lambda_2^{(i)}(M^{(i+1)} - Y^{(i+1)})$
    $\lambda_1^{(i+1)} \leftarrow \gamma\lambda_1^{(i)}, \lambda_2^{(i+1)} \leftarrow \gamma\lambda_2^{(i)}, i \leftarrow i + 1$;
  untile convergence or reach the maximum iteration times.
```

4. Experiments

4.1. Experiment setup

We have conducted several experiments on two mocap datasets to show the effectiveness of our method. The first one is the CMU mocap dataset[5] which contains a huge collection of mocap data. We pick out 12 motion sequences from 8 subjects[6] including multiple types of actions such as walk, jump, run, boxing, and tai chi. Because data from CMU dataset are very clean and complete [37], we used them in the synthetic experiments. In order to test for real applications, we captured 3 long motion sequences using Motion Analysis Eagle-4 Digital RealTime System[7] and these data consists of three daily actions (i.e., walk, run and jump) and the total frame number is 3178. Different from the CMU dataset, our own captured motion sequences naturally contains incomplete and noisy data as shown in Fig. 1.

Our method does not need the support of databases. To evaluate the performance of our method, we compare it with the other four methods: Linear interpolation (Linear), Spline interpolation (Spline), Dynammo [30] and SVT [23]. The first two are widely used in practice and the third and fourth are two state-of-the-art methods. For Linear and Spline, we combine them with a Gaussian filter to handle the de-noising problem. In particular, we first use the Gaussian filter to remove the noise and then use a simple threshold method to find out the clear entries, and finally apply these two interpolation methods based on the detected entries to predict and correct the noise entries. This way, all methods can handle both of the two subproblems of the mocap data refinement problem and can make a fair comparison. To quantify the refined results, following the work [4,19,30,47,51,58], the Root Mean Squared Error (RMSE) measurement is adopted:

$$\mathrm{RMSE}(f_i, \hat{f}_i) = \sqrt{\frac{1}{n_p}\|f_i - \hat{f}_i\|^2}, \tag{27}$$

where f_i is the imperfect pose, $\hat{f}_i$ is the refined pose and n_p is the total number of imperfect entries (i.e., missing and noise entries) in f_i.

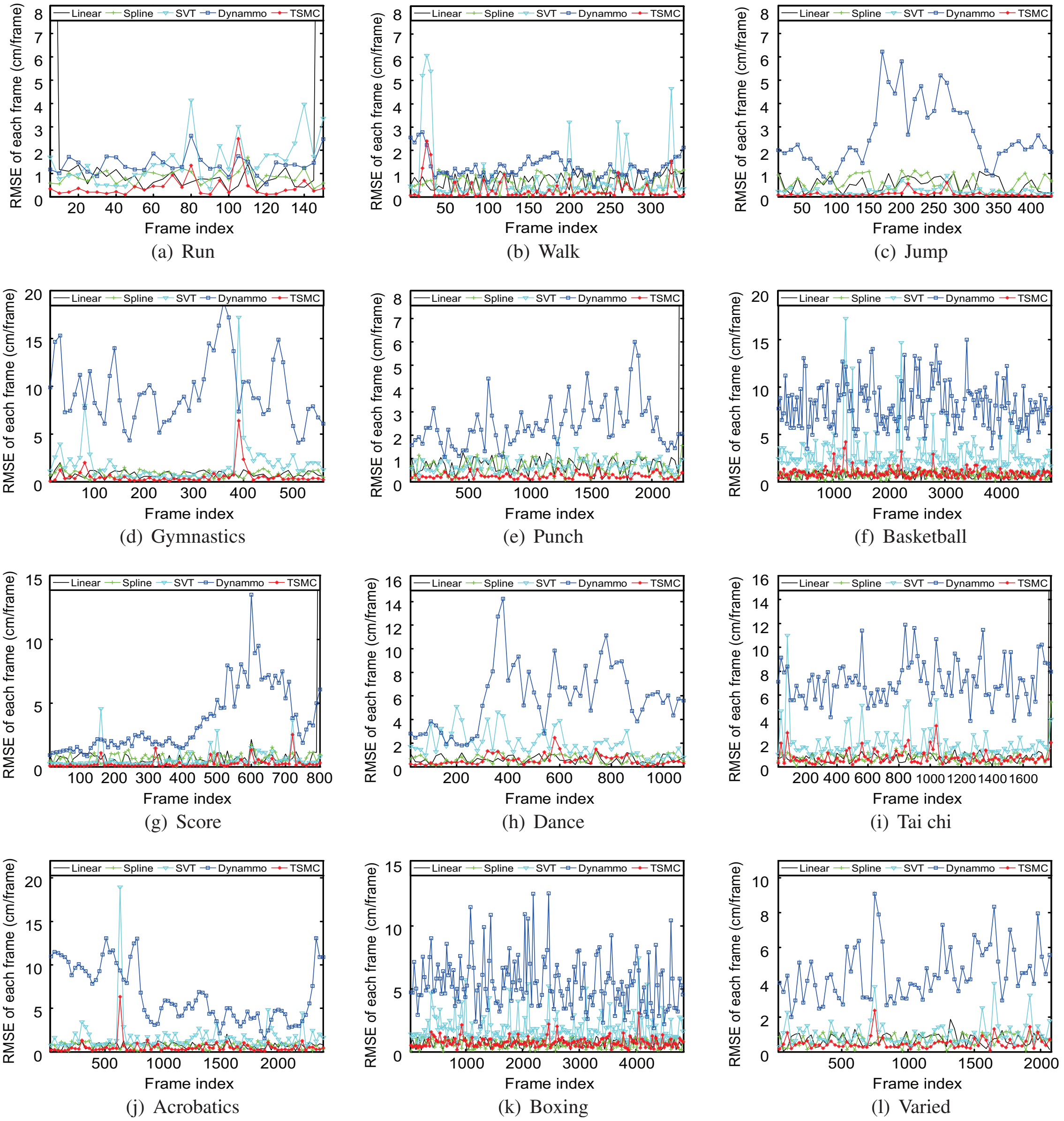
4.2. Evaluation on synthetic data

Since it is at most about 30–40% of the data is missing or noisy in practice [29], we fix the ratio of the missing or noisy data to 30–40% to evaluate all the algorithms. Using the selected 12 motion sequences, we systematically simulate four classical situations to synthesize four different kinds of corrupted data, which are listed as follows.

[5] http://mocap.cs.cmu.edu/.

[6] The selected subjects include 2, 5, 6, 9, 10, 12, 13, 49.

[7] http://en.souvr.com/product/200908/2530.html.



(a) Run (b) Walk (c) Jump

(d) Gymnastics (e) Punch (f) Basketball

(g) Score (h) Dance (i) Tai chi

(j) Acrobatics (k) Boxing (l) Varied

Fig. 6. Prediction results using different methods with 40% data are randomly missing.

Randomly lose data. In this case, we randomly removed 40% data from each sequence, so that both the missing markers number and missing time are random.

Regularly lose data. Different from the first case, we regularly removed about 30% data wherein the number of selected missing markers is fixed to 10 for each incomplete frame and each selected markers miss 60 frames. Thus some long gaps appear in the motion sequence under such condition.

Randomly corrupt data. To evaluate the de-noising capability, we randomly added gaussian noise ($\sigma = 2$) on 30% data for each motion sequence.

Mixed corrupt data. In this case, we randomly remove 30% data and then corrupt 30% of the observed part data with gaussian noise ($\sigma = 2$), so these data can be used to simultaneously evaluate the prediction and de-noising capability of all the methods.

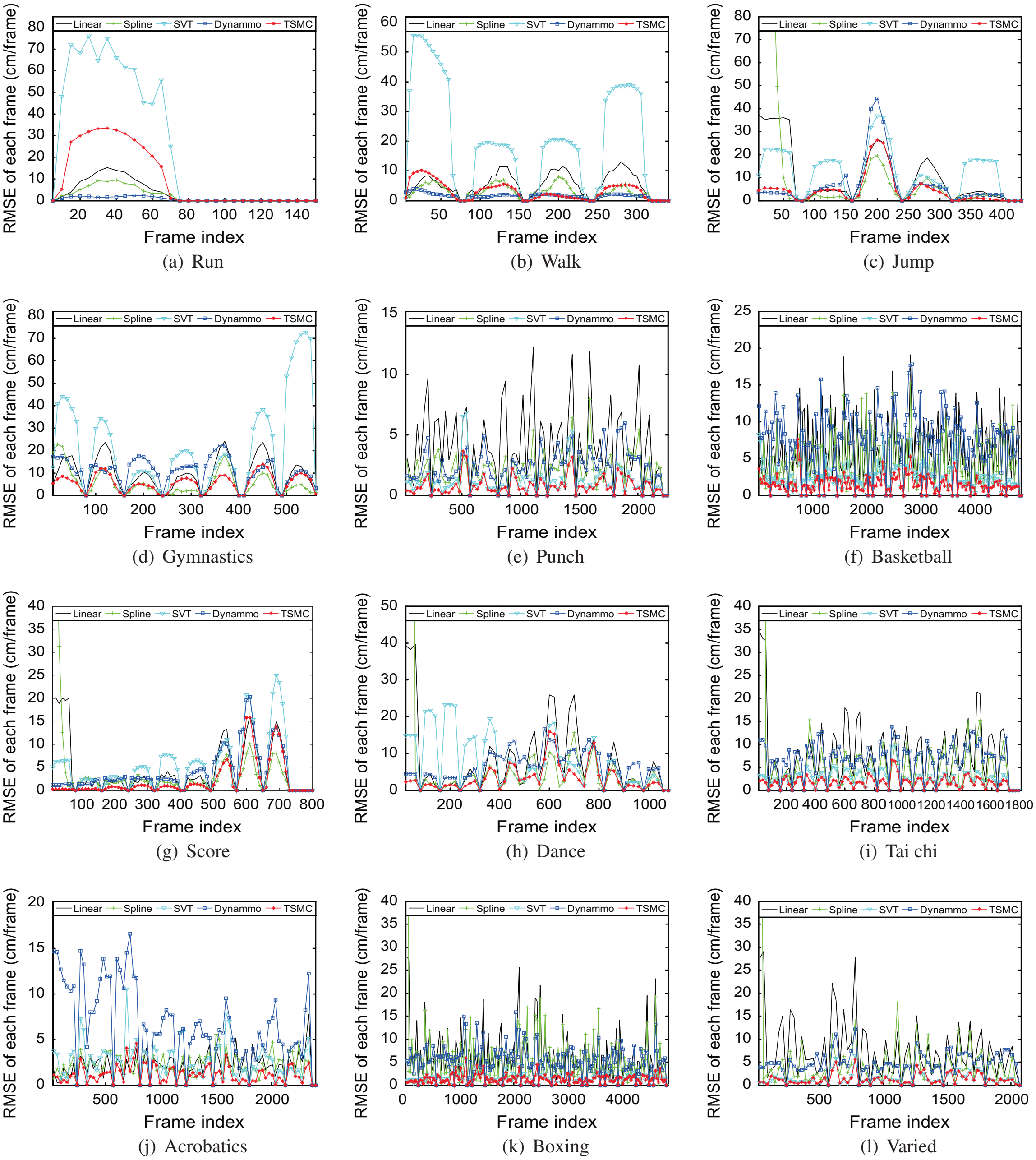
Note that we add noise on the original motion data recorded in the ASF\AMC[8] files, so that when we convert the unit of measurement into centimeter, we multiple the results by 5.6444.[9] We use the above generated data to evaluate the prediction

[8] http://research.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/ASF-AMC.html.

[9] http://mocap.cs.cmu.edu/faqs.php.

and de-noising capability of each algorithm. In order to investigate the proposed trust data detection method, we compare the performance of TSMC with and without the step of trust data detection using the mixed corrupted data. The experimental results are shown in Figs. 6–10 one by one. From these evaluation results, we obtain the following conclusions:

- From Figs. 6–9, we can see that our method outperforms all competitors not only in predicting missing values but also in de-noising most of the time. More importantly, the variance of RMSE obtained by our proposed TSMC is relatively small, which means that the outcomes of our method are very stable. We believe the reason is that our model does not only exploit the low-rank structure, but also the temporal stability property. Moreover, the trust data detection method boosts the performance of our method as shown in Fig. 10.



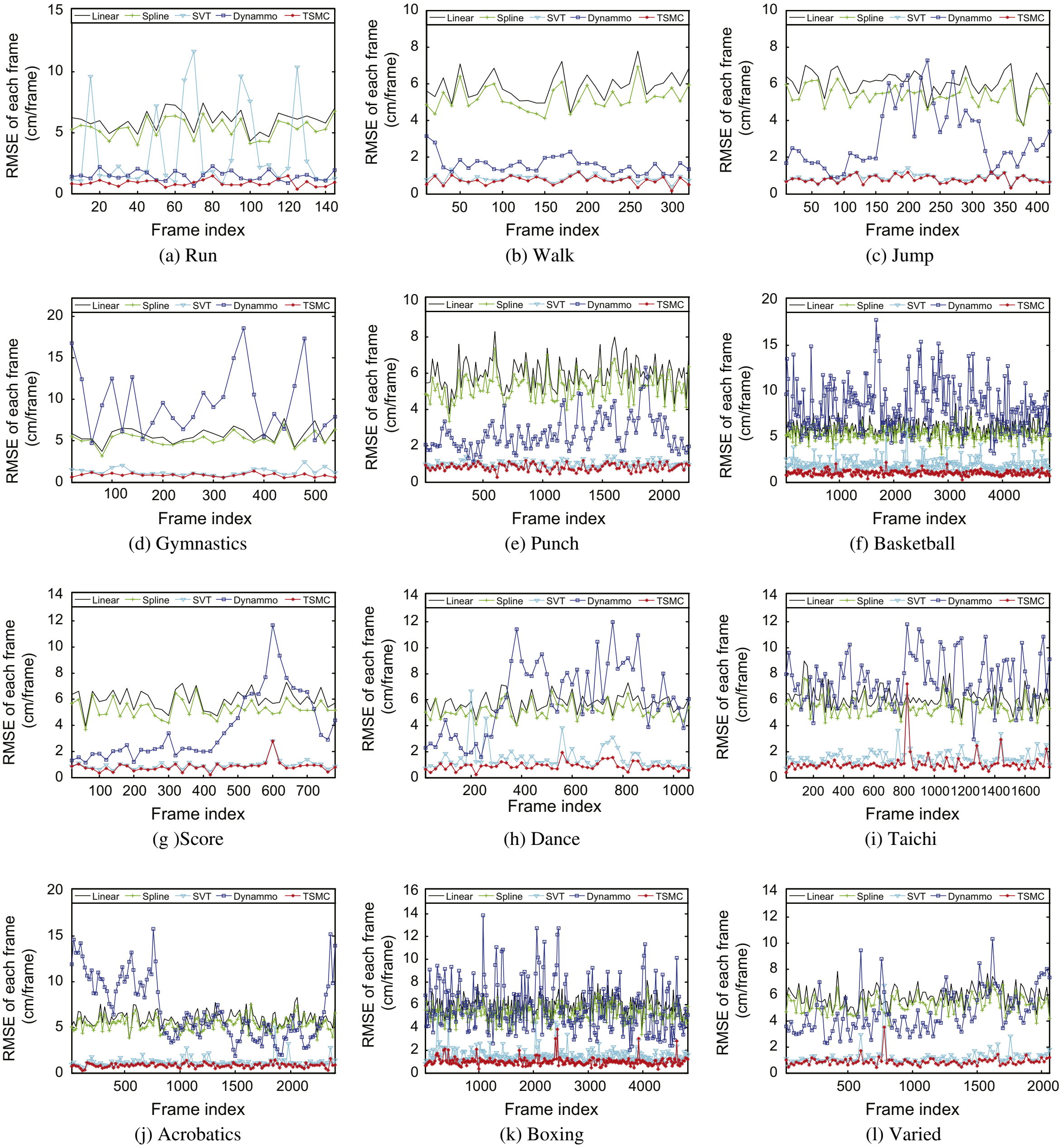
(a) Run (b) Walk (c) Jump

(d) Gymnastics (e) Punch (f) Basketball

(g) Score (h) Dance (i) Tai chi

(j) Acrobatics (k) Boxing (l) Varied

Fig. 7. Prediction results using different methods with about 30% data are regularly missing wherein the number of missing markers is 10 and missing time is 60 frames in each time.



(a) Run (b) Walk (c) Jump

(d) Gymnastics (e) Punch (f) Basketball

(g)Score (h) Dance (i) Taichi

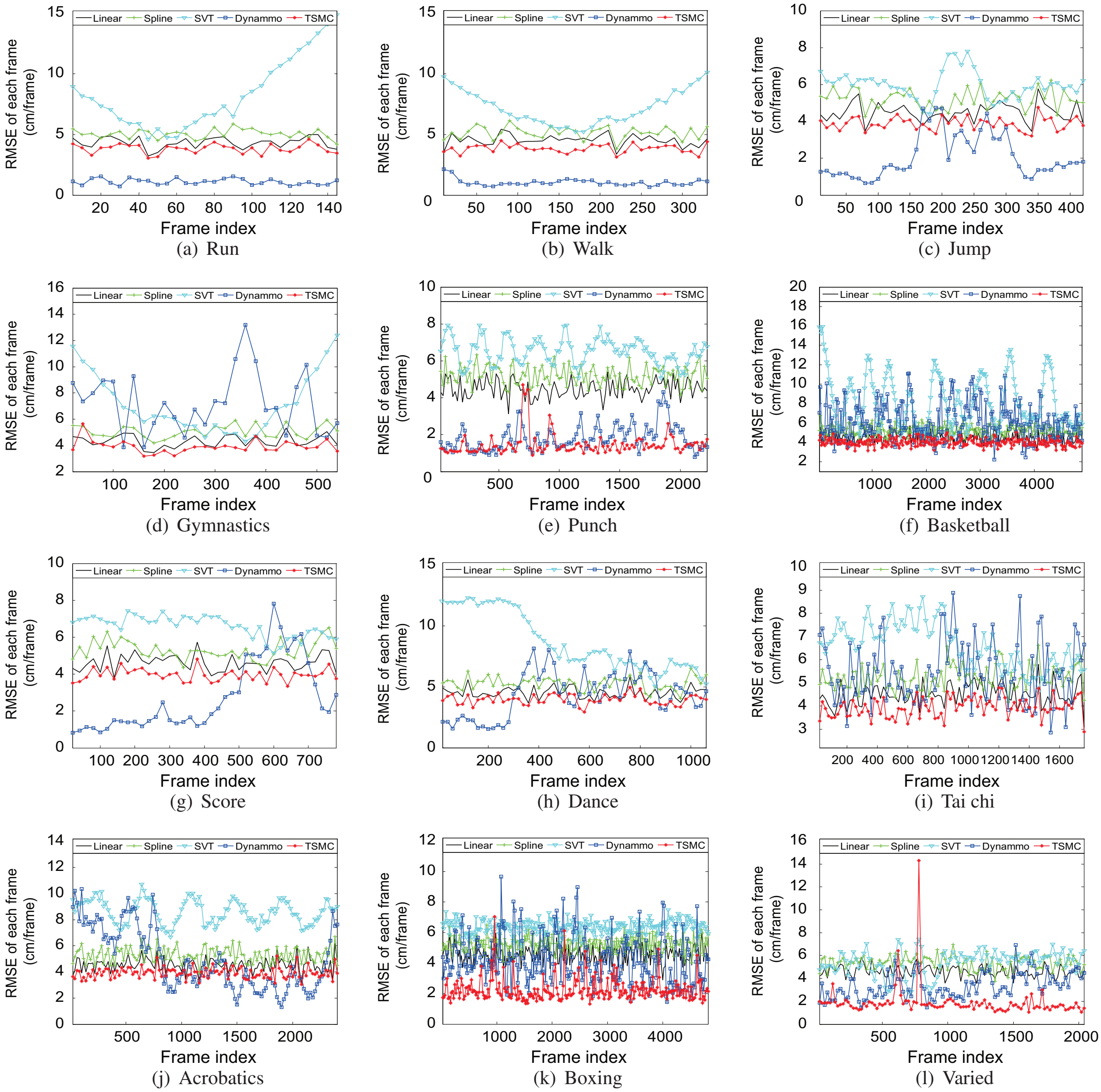
(j) Acrobatics (k) Boxing (l) Varied

Fig. 8. De-noising results using different methods with 30% data are randomly noised.

- SVT [23] and our method are much more suitable to handle long sequences (e.g., boxing and tai chi sequences in Fig. 7(k) and (i)) than short sequences (e.g., run and walk sequences in Fig. 7(a) and (b)).
- Linear and Spline methods are suitable for handling short time randomly missing data as Fig. 6. However they are unable to handle long time data missing as Fig. 7 and the special case that the missing data appears at the end of a sequence as Fig. 7(g)–(i).
- Dynammo [30] works very well in handling periodic actions such as walking and running. However, it will corrupt when the noise increases or some data are randomly missing as Figs. 6 and 8.

4.3. Evaluation on real data

We also have evaluated all of these methods using our own captured three long sequences of mocap data. In these mocap data, some markers are missing for a long period of time and what is more serious is that some missing markers appear at



(a) Run (b) Walk (c) Jump

(d) Gymnastics (e) Punch (f) Basketball

(g) Score (h) Dance (i) Tai chi

(j) Acrobatics (k) Boxing (l) Varied

Fig. 9. Prediction and de-noising using different methods with 30% data are randomly missing and 30% of the observed data are corrupted by gaussian noise ($\sigma = 2$).

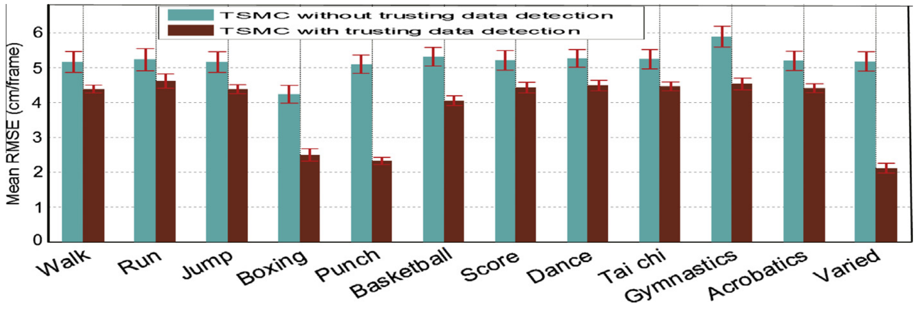
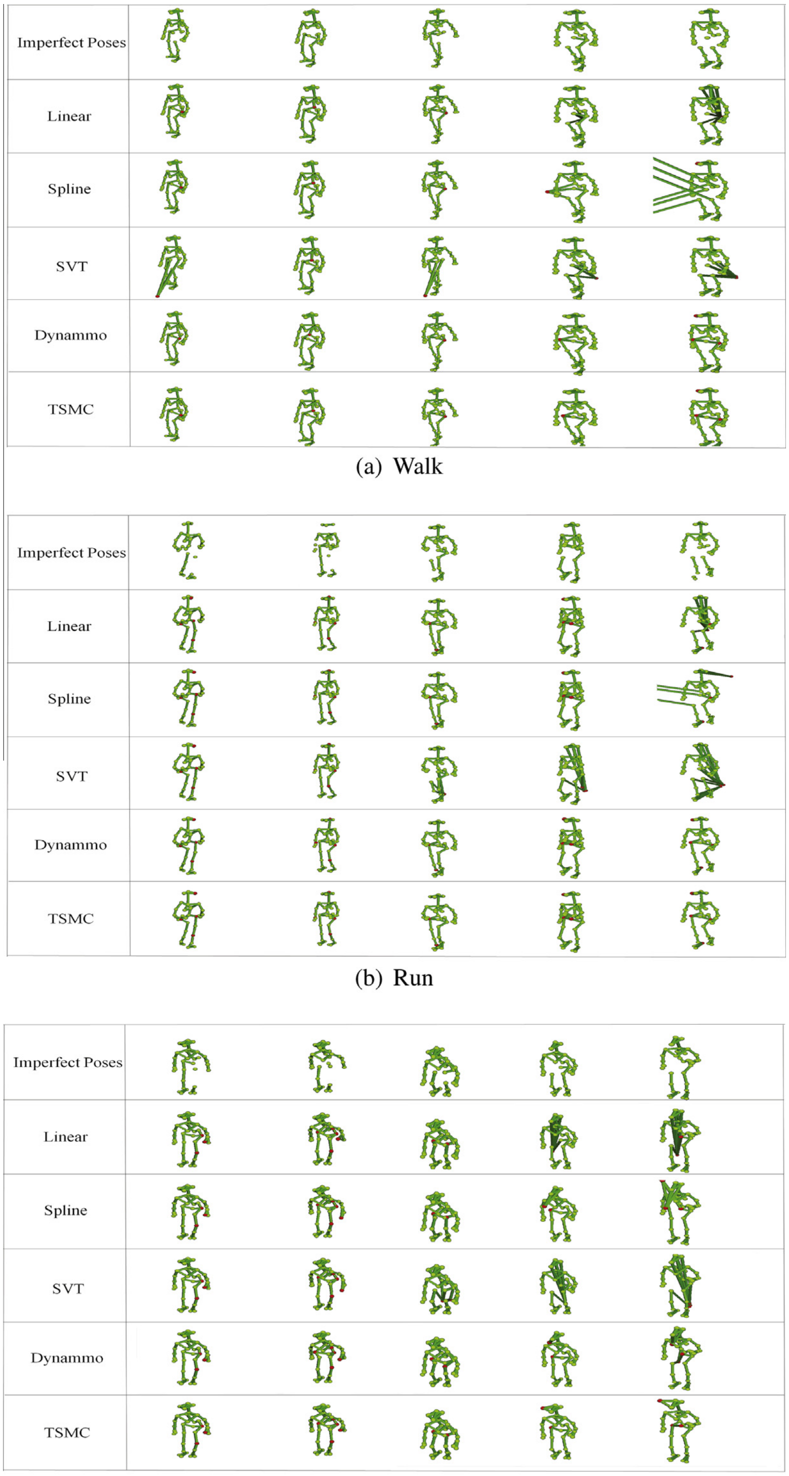


Fig. 10. Performance comparison of our proposed TSMC algorithm with and without the step of trust data detection using mixed corrupted data.

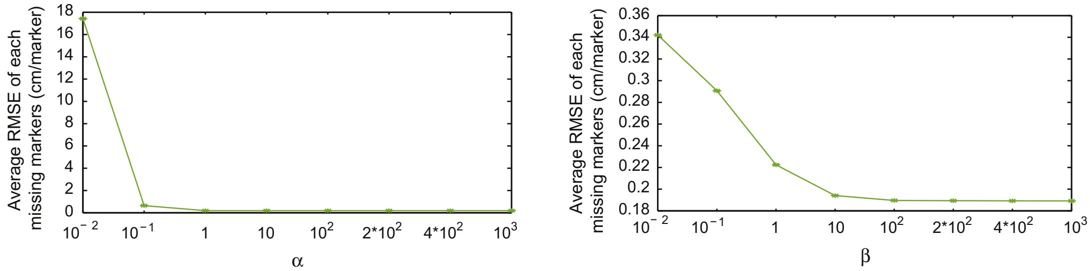


(a) Walk

(b) Run

(c) Jump

Fig. 11. Performance comparison results of different algorithms in our three motion sequences. In each subgraph, the first row is the raw incomplete poses and the next rows are the refined results using Linear, Spline, SVT [23], Dynammo [30] and our method respectively. The red points represent the predicted missing markers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Evaluate α with$\beta = 10^2$

(b) Evaluate β with $\alpha = 1$

Fig. 12. Performance variance w.r.t. α and β.

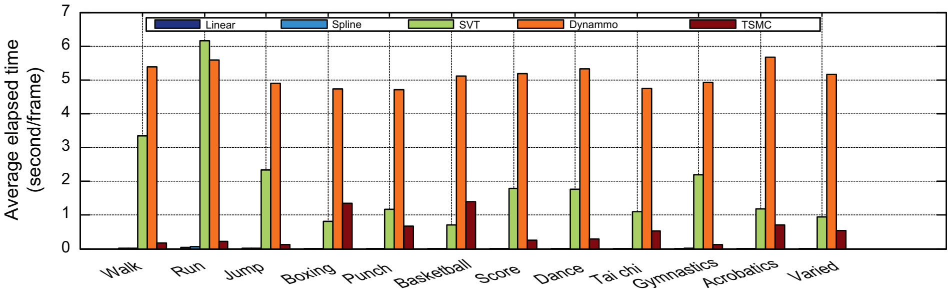


Fig. 13. The elapsed time comparison between different methods.

the end. Fig. 11 shows the refinement results using different methods. As mentioned above, Spline and Linear methods are only suitable for short-time data missing so they are easy to fail under such a condition. When the missing markers appear at the end, both linear and spline method are unable to correctly predict the missing values as shown in Fig. 11. Some markers predicted by these two methods deviate significantly from the correct positions due to lack of control points at the end of the motion data. SVT [23] only exploits the low-rank structure of motion data and cannot handle the cases where markers are missing for a long period of time. From the last two columns of each sub-image in Fig. 11, the shortcoming of SVT [23] is shown that it is unable to correctly predict some missing values. Relatively speaking, we find that Dynammo [30] offers the best performance for all the three motion sequences. However, when some markers are missing for a long period of time, Dynammo [30] also fails to correctly predict the missing values. As shown in the last sub-image of Fig. 11(c), one head marker predicted by Dynammo [30] drifts a little away due to the reason that this marker is missing for a long period of time at the end of the motion data. With exploiting both the low-rank structure and temporal stability of motion data, our method works demonstrates the robustness on all three real sequences.

4.4. Parameter sensitivity and running time study

We investigate the parameter sensitivity of α and β in our model using a long sequence with multiple actions and tune these two parameters from 10^{-2} to 10^3. From Fig. 12 we see that α and β are insensitive for a large range. Thus we simply set $\alpha = 1$ and $\beta = 10^2$ in our experiments. More importantly, our method is faster than Dynammo [30] and SVT [23] most of the time as shown in Fig. 13. Meanwhile, one can notice that our method takes up more time than SVT [23] in two human activities: boxing and basketball. This may appear contradicting to our early claim that our method was much faster and robust than SVT [23]. To demystify this inconsistency, we find that SVT [30] convergences too early in handling a large-scale imperfect human motion matrix in these two cases where the total frame numbers are 4840 and 4905. Therefore in Fig. 9(k) and (f), SVT [30] is unable to correctly refine such a large-scale motion data, as opposed to our method, as shown in Fig. 9(k) and (f), which demonstrates the robustness of our method.

5. Conclusion

Motion capture data typically consists of imperfection elements, such as noise and missing makers. Cleaning up the imperfect data is known as the mocap data refinement. A lot of research efforts have been made to automate this process. In this paper we have made the following contributions towards solving the problem: presented a new method making full use of both the low-rank structure and temporal stability properties of the motion data and convert it into a matrix completion problem; developed a fast optimization method derived from the ALM algorithm to solve the proposed model;

presented an efficient trust data detection method to automate data processing and boost the computation performance. Extensive experiments on both synthetic and real data have demonstrated the effectiveness of our proposed technique.

However, in the current development, the foot contact with the ground has not been taken into account, which may sometimes lead to feet sliding. Fortunately, this issue has been well investigated [22,25,46]. Since human motion data contains strong structural information, we would like to incorporate the human skeleton information into our proposed model in the future. In addition, techniques such as key-frame reduction [35,40,41,59] and motion data segmentation techniques [36,61] allow human motion data to be dealt with considering the specific features, which can be used with our framework to further improve the speed of our algorithm.

Acknowledgements

This research is supported by the National High Technology Research and Development Program (2012*AA*011502), the National Key Technology R&D Program (2013*BAH*59*F*00) and the Zhejiang Provincial Natural Science Foundation of China (*LY*13*F*020001), and partially supported by the grant of the "Sino-UK Higher Education Research Partnership for Ph.D. Studies" Project funded by the Department of Business, Innovation and Skills of the British Government and Ministry of Education of P.R. China. The authors would like to thank Ranch YQ Lai, Shusen Wang and Zhihua Zhang for sharing their source code and providing helpful comments on this manuscript. In addition, the authors would like to thank editors and reviewers for their constructive comments on this manuscript.

Appendix A. Derivation of the updating rules for our Algorithm 2

Before introducing how to derivate the updating rules for our Algorithm 2, we give an useful theorem from some literatures [7,33,34,53] as follows.

Theorem 1. For any $\tau \geqslant 0, A, B \in \mathbb{R}^{m\times n}, \Omega \in \{0,1\}^{m\times n}, \mathcal{D}_\tau$ and $\mathcal{S}_\tau$ defined as in Eqs. (25) and (26), we have

$$\mathcal{D}_\tau(B) = \arg\min_A \ \tau\|A\|_1 + \frac{1}{2}\|A-B\|_F^2, \tag{A.1}$$

$$\mathcal{S}_\tau(B) = \arg\min_A \ \tau\|A\|_* + \frac{1}{2}\|A-B\|_F^2, \tag{A.2}$$

$$\widehat{A} = \arg\min_A \ \tau\|A\circ\Omega\|_1 + \frac{1}{2}\|A-B\|_F^2, \tag{A.3}$$

where $\widehat{A} = \Omega\circ\mathcal{D}_\tau(B) + \overline{\Omega}\circ B$.

By the spirit of ALM, we rewrite Eq. (21) to its corresponding augmented Lagrange function:

$$\begin{aligned}\mathcal{J}(Y,E,M,Y_1,Y_2) = {} & \|Y\|_* + \alpha\|\Omega\circ E\|_1 + \frac{\beta}{2}\|MO\|_F^2 + \langle Y_1, X-Y-E\rangle + \frac{\lambda_1}{2}\|X-Y-E\|_F^2 \\ & + \langle Y_2, M-Y\rangle + \frac{\lambda_2}{2}\|Y-M\|_F^2.\end{aligned} \tag{A.4}$$

Given the initial setting $Y_1^{(0)} = Y_2^{(0)} = X/\max(\|X\|_2, \|X\|_\infty)$, $Y^{(0)} = \mathbf{0}, E^{(0)} = \mathbf{0}$ and $M^{(0)} = \mathbf{0}$, the optimization problem Eq. (A.4) can be solved via the following steps.

Computing $Y^{(i+1)}$: Fix $E^{(i)}, M^{(i)}$, $Y_1^{(i)}$ and $Y_2^{(i)}$, and minimize $\mathcal{J}(Y, E^{(i)}, M^{(i)}, Y_1^{(i)}, Y_2^{(i)})$ for $Y^{(i+1)}$:

$$Y^{(i+1)} = \arg\min_Y \ \|Y\|_* + \langle Y_1^{(i)}, X-Y-E^{(i)}\rangle + \frac{\lambda_1}{2}\|X-Y-E^{(i)}\|_F^2 + \langle Y_2^{(i)}, M^{(i)}-Y\rangle + \frac{\lambda_2}{2}\|Y-M^{(i)}\|_F^2, \tag{A.5}$$

$$\Rightarrow Y^{(i+1)} = \arg\min_Y \ \|Y\|_* + \frac{\lambda_1+\lambda_2}{2}\left\|Y - \frac{Y_1^{(i)} + Y_2^{(i)} + \lambda_1(X-E^{(i)}) + \lambda_2 M^{(i)}}{\lambda_1+\lambda_2}\right\|_F^2. \tag{A.6}$$

Based on Eq. (A.2), we can solve the above Eq. (A.5) as

$$Y^{(i+1)} = \mathcal{S}_{1/(\lambda_1+\lambda_2)}(Z_Y), \tag{A.7}$$

where $Z_Y = \dfrac{Y_1^{(i)} + Y_2^{(i)} + \lambda_1(X-E^{(i)}) + \lambda_2 M^{(i)}}{\lambda_1+\lambda_2}$.

Computing $E^{(i+1)}$: Fix $Y^{(i+1)}, M^{(i)}, Y_1^{(i)}$ and $Y_2^{(i)}$ to compute $E^{(i+1)}$ as follows:

$$E^{(i+1)} = \arg\min_E \ \alpha\|\Omega\circ E\|_1 + \langle Y_1^{(i)}, X-Y^{(i+1)}-E\rangle + \frac{\lambda_1}{2}\|X-Y^{(i+1)}-E\|_F^2, \tag{A.8}$$

$$\Rightarrow E^{(i+1)} = \arg\min_E \ \alpha\|\Omega\circ E\|_1 + \frac{\lambda_1}{2}\left\|E - \left(\frac{1}{\lambda_1}Y_1^{(i)} + X - Y^{(i+1)}\right)\right\|_F^2. \tag{A.9}$$

and we get the updating rule for $E^{(i+1)}$ according to Eq. (A.3) as follows

$$E^{(i+1)} = \Omega \circ \mathcal{D}_{\alpha/\lambda_1}(Z_e) + \hat{\Omega} \circ Z_e, \tag{A.10}$$

where $Z_e = \frac{1}{\lambda_1} Y_1^{(i)} + X - Y^{(i+1)}$.

Computing $M^{(i+1)}$: Fix $Y^{(i+1)}, E^{(i+1)}, Y_1^{(i)}$ and $Y_2^{(i)}$ then calculate $M^{(i+1)}$ as follows:

$$M^{(i+1)} = \arg\min_M \frac{\beta}{2}\|MO\|_F^2 + \langle Y_2^{(i)}, M - Y^{(i+1)}\rangle + \frac{\lambda_2}{2}\|Y^{(i+1)} - M\|_F^2. \tag{A.11}$$

The derivative of the above Eq. (A.11) w.r.t. M is

$$\frac{\partial \mathcal{J}}{\partial M} = M(\beta O O^T + \lambda_2 I) + Y_2^{(i)} - \lambda_2 Y^{(i+1)}. \tag{A.12}$$

We can get the optimal value of M by setting Eq. (A.12) to zero and the optimal value of M is

$$M^{(i+1)} = \left(\lambda_2 Y^{(i+1)} - Y_2^{(i)}\right)(\beta O O^T + \lambda_2 I)^{-1}. \tag{A.13}$$

Computing $Y_1^{(i+1)}$ **and** $Y_2^{(i+1)}$: Fixing $Y^{(i+1)}, E^{(i+1)}$ and $M^{(i+1)}$, we calculate $Y_1^{(i+1)}$ and $Y_2^{(i+1)}$ as follows [6,7,33,53]:

$$Y_1^{(i+1)} = Y_1^{(i)} + \lambda_1(X - Y^{(i+1)} - E^{(i+1)}), \tag{A.14}$$

$$Y_2^{(i+1)} = Y_2^{(i)} + \lambda_2(M^{(i+1)} - Y^{(i+1)}). \tag{A.15}$$

Similarly, we also update λ_1 and λ_2 with a positive scalar $\gamma > 1$:

$$\lambda_1^{(i+1)} = \gamma\lambda_1^{(i)}, \quad \lambda_2^{(i+1)} = \gamma\lambda_2^{(i)}, \tag{A.16}$$

so that $\{\lambda_1^{(i)}\}$ and $\{\lambda_2^{(i)}\}$ are two increasing sequences and the ALM will converge to the optimal solution as proved in [6,33].

Till now, we get all of the updating rules of $Y, E, M, Y_1, Y_2, \lambda_1$ and λ_2. The optimization method is summarized in Algorithm 2.

Appendix B. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.ins.2014.03.013.

References

[1] A. Aristidou, J. Cameron, J. Lasenby, Real-time estimation of missing markers in human motion capture, in: Proceedings of the 2nd International Conference on Bioinformatics and Biomedical Engineering (ICBBE), IEEE, 2008, pp. 1343–1346.

[2] A. Aristidou, J. Lasenby, Real-time marker prediction and CoR estimation in optical motion capture, Visual Comput. 29 (1) (2013) 7–26.

[3] J. Barca, G. Rumantir, R. Li, Noise filtering of new motion capture markers using modified k-means, Comput. Intell. Multimedia Process.: Recent Adv. 96 (2008) 167–189.

[4] J. Baumann, B. Krüger, A. Zinke, A. Weber, Data-driven completion of motion capture data, in: Proceedings of Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS), Eurographics Association, 2011, pp. 111–118.

[5] J. Baumann, B. Krüger, A. Zinke, A. Weber, Filling long-time gaps of motion capture data, in: Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA), vol. 33, ACM, Vancouver, Canada, 2011b.

[6] D.P. Bertsekas, Nonlinear Programming, Athena Scientific, 1999.

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

[8] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, Found. Comput. Math. 9 (6) (2009) 717–772.

[9] C.-W. Chun, O.C. Jenkins, M.J. Mataric, Markerless kinematic model and motion capture from volume sequences, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, 2003, pp. 475–482.

[10] P. Craven, G. Wahba, Smoothing noisy data with spline functions, Numer. Math. 31 (4) (1979) 377–403.

[11] D.L. Donoho, For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution, Commun. Pure Appl. Math. 59 (2006) 797–829.

[12] P.A. Federolf, A novel approach to solve the missing marker problem in marker-based motion analysis that exploits the segment coordination patterns in multi-limb motion data, PloS one 8 (10) (2013) 1–13.

[13] V. Ganapathi, C. Plagemann, D. Koller, S. Thrun, Real time motion capture using a single time-of-flight camera, in: Proceedings of I EEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 755–762.

[14] D. Garcia, Robust smoothing of gridded data in one and higher dimensions with missing values, Comput. Stat. Data Anal. 54 (4) (2010) 1167–1178.

[15] G.H. Golub, C.F.V. Loan, Matrix Computations, vol. 1, The Johns Hopkins University Press, 1996.

[16] R.M. Heiberger, R.A. Bencker, Design of an s function for robust regression using iteratively reweighted least squares, J. Comput. Graph. Stat. 1 (3) (1992) 181–196.

[17] L. Herda, P. Fua, R. Plankers, R. Boulic, D. Thalmann, Skeleton-based motion capture for robust reconstruction of human motion, in: Proceedings of Computer Animation, IEEE, 2000, pp. 77–83.

[18] L. Herda, P. Fua, R. Plänkers, R. Boulic, D. Thalmann, Using skeleton-based tracking to increase the reliability of optical motion capture, Hum. Movement Sci. 20 (3) (2001) 313–341.

[19] J. Hou, L-P. Chau, Y. He, J. Chen, N. Magnenat-Thalmann, Human motion capture data recovery via trajectory-based sparse representation, in: Proceedings of IEEE International Conference on Image Processing (ICIP), IEEE, 2013, pp. 709–713.

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

[21] A.R. Jensenius, K. Nymoen, S.A. Skogstad, A. Voldsund, A study of the noise-level in two infrared marker-based motion capture system, in: Proceedings of the 9th Sound and Music Computing Conference (SMC), Aalborg University Press, Copenhagen, Denmark, 2012, pp. 258-263.

[22] L. Kovar, J. Schreiner, M. Gleicher, Footskate cleanup for motion capture editing, in: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM, 2002, pp. 97–104.

[23] R.Y. Lai, P.C. Yuen, K.K. Lee, Motion capture data completion and denoising by singular value thresholding, in: Proceedings of Eurographics, Eurographics Association, 2011, pp. 45–48.

[24] B. Le Callennec, R. Boulic, Robust kinematic constraint detection for motion data, in: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Eurographics Association, 2006, pp. 281–290.

[25] J. Lee, K.H. Lee, Precomputing avatar behavior from human motion data, in: Proceedings of the 2004 Eurographics/SIGGRAPH Symposium on Computer Animation, The Eurographics Association, 2004, pp. 79–87.

[26] J. Lee, S.Y. Shin, Motion fairing, in: Proceedings of Computer Animation, IEEE, 1996, pp. 136–143.

[27] J. Lee, S.Y. Shin, General construction of time-domain filters for orientation data, IEEE Trans. Visual. Comput. Graph. 8 (2) (2002) 119–128.

[28] T.C. Lee, Smoothing parameter selection for smoothing splines: a simulation study, Comput. Stat. Data Anal. 42 (1–2) (2003) 139–148.

[29] S. Lemercier, M. Moreau, M. Moussaèd, G. Theraulaz, S. Donikian, J. Pettrè, Reconstructing motion capture data for human crowd study, Motion Games (2011) 365–376.

[30] 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, 2009, pp. 507–516.

[31] 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.

[32] S. Li, Fast Algorithms for Sparse Matrix Inverse Computations, Ph.D. Thesis, Stanford University, 2009.

[33] Z. Lin, M. Chen, L. Wu, Y. Ma, The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-rank Matrices, Technical Report UILU-ENG-09-2215, UIUC, October 2009, 2010.

[34] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, Y. Ma, Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix, in: Proceedings of International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), Citeseer, 2009.

[35] X. Liu, A. Hao, D. Zhao, Optimization-based key frame extraction for motion capture animation, Visual Comput. 29 (1) (2013) 85–95.

[36] A. López-Méndez, J. Gall, J.R. Casas, L.J. Van Gool, Metric learning from poses for temporal clustering of human motion, in: Proceedings of the 23rd British Machine Vision Conference (BMVC), BMVA Press, 2012, pp. 1–12.

[37] H. Lou, J. Chai, Example-based human motion denoising, IEEE Trans. Visual. Comput. Graph. 16 (5) (2010) 870–879.

[38] T.B. Moeslund, E. Granum, A survey of computer vision-based human motion capture, Comput. Vision Image Understan. 81 (3) (2001) 231–268.

[39] T.B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, Comput. Vision Image Understan. 104 (2) (2006) 90–126.

[40] O. Önder, Ç. Erdem, T. Erdem, U. Güdükbay, B. Özgüç, Combined filtering and key-frame reduction of motion capture data with application to 3DTV, in: Proceedings of the 14th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media (WSCG'2006), Union Agency, 2006, pp. 29–30.

[41] O. Önder, U. Güdükbay, B. Özgüç, T. Erdem, C. Erdem, M. Özkan, Keyframe reduction techniques for motion capture data, in: Proceedings of 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, IEEE, 2008, pp. 293–296.

[42] R. Poppe, Vision-based human motion analysis: an overview, Comput. Vision Image Understan. 108 (1) (2007) 4–18.

[43] S. Rallapalli, L. Qiu, Y. Zhang, Y.-C. Chen, Exploiting temporal stability and low-rank structure for localization in mobile networks, in: Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking, ACM, 2010, pp. 161–172.

[44] L. Ren, A. Patrick, A.A. Efros, J.K. Hodgins, J.M. Rehg, A data-driven approach to quantifying natural human motion, ACM Trans. Graph. 24 (3) (2005) 1090–1097.

[45] H.P. Shum, E.S. Ho, Y. Jiang, S. Takagi, Real-time posture reconstruction for microsoft kinect, IEEE Trans. Cybernet. 43 (5) (2013) 1357–1369.

[46] K.W. Sok, M. Kim, J. Lee, Simulating biped behaviors from human motion data, ACM Trans. Graph. 26 (107) (2007) 80–89.

[47] C.-H. Tan, J. Hou, L.-P. Chau, Human motion capture data recovery using trajectory-based matrix completion, Electron. Lett. 49 (12) (2013) 752–754.

[48] J. Tautges, A. Zinke, B. Krüger, J. Baumann, A. Weber, T. Helten, M. Müller, H.-P. Seidel, B. Eberhardt, Motion reconstruction using sparse accelerometer data, ACM Trans. Graph. 30 (3) (2011) 18–29.

[49] A. Ude, C.G. Atkeson, M. Riley, Planning of joint trajectories for humanoid robots using b-spline wavelets, Proceedings of IEEE International Conference on Robotics and Automation(ICRA), vol. 3, IEEE, 2000, pp. 2223–2228.

[50] M. Wall, A. Rechtsteiner, L. Rocha, Singular value decomposition and principal component analysis, Pract. Approach Microarray Data Anal. (2003) 91–109.

[51] 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.

[52] P. Wang, Z. Pan, M. Zhang, R.W. Lau, H. Song, The alpha parallelogram predictor: a lossless compression method for motion capture data, Inform. Sci. 232 (2013) 1–10.

[53] S. Wang, Z. Zhang, Colorization by matrix completion, in: Proceedings of the Twenty-Sixth National Conference on Artificial Intelligence (AAAI), AAAI Press, 2012.

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

[55] J. Wright, Y. Peng, Y. Ma, A. Ganesh, S. Rao, Robust principal component analysis: exact recovery of corrupted low-rank matrices via convex optimization, in: Proceedings of Advances in Neural Information Processing Systems (NIPS), Citeseer, 2009.

[56] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.

[57] Q. Wu, P. Boulanger, Real-time estimation of missing markers for reconstruction of human motion, in: Proceedings of the 2011 XIII Symposium on Virtual Reality (SVR), IEEE, 2011, pp. 161–168.

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

[59] Q. Zhang, X. Xue, D. Zhou, X. Wei, Motion key-frames extraction based on amplitude of distance characteristic curve, Int. J. Comput. Intell. Syst. (ahead-of-print) (2013) 1–9.

[60] Z. Zhang, Microsoft kinect sensor and its effect, IEEE Multimedia 19 (2) (2012) 4–10.

[61] F. Zhou, F. De la Torre, J.K. Hodgins, Hierarchical aligned cluster analysis for temporal clustering of human motion, IEEE Trans. Pattern Anal. Mach. Intell. 35 (3) (2013) 582–596.