Low-latency compression of mocap data using learned spatial decorrelation transform

Junhui Hou [a], Lap-Pui Chau [a], Nadia Magnenat-Thalmann [b], Ying He [c,*]

[a] *School of Electrical and Electronics Engineering, Nanyang Technological University, 639798, Singapore*
[b] *Institute for Media Innovation, Nanyang Technological University, 639798, Singapore*
[c] *School of Computer Engineering, Nanyang Technological University, 639798, Singapore*

ARTICLE INFO

ABSTRACT

Due to the growing needs of motion capture (mocap) in movie, video games, sports, etc., it is highly desired to compress mocap data for efficient storage and transmission. Unfortunately, the existing compression methods have either high latency or poor compression performance, making them less appealing for time-critical applications and/or network with limited bandwidth. This paper presents two efficient methods to compress mocap data with low latency. The first method processes the data in a frame-by-frame manner so that it is ideal for mocap data streaming. The second one is clip-oriented and provides a flexible trade-off between latency and compression performance. It can achieve higher compression performance while keeping the latency fairly low and controllable. Observing that mocap data exhibits some unique spatial characteristics, we learn an orthogonal transform to reduce the spatial redundancy. We formulate the learning problem as the least square of reconstruction error regularized by orthogonality and sparsity, and solve it via alternating iteration. We also adopt a predictive coding and temporal DCT for temporal decorrelation in the frame- and clip-oriented methods, respectively. Experimental results show that the proposed methods can produce higher compression performance at lower computational cost and latency than the state-of-the-art methods. Moreover, our methods are general and applicable to various types of mocap data.

1. Introduction

As a highly successful technique, motion capture (mocap) has been widely used to animate virtual characters in distributed virtual reality applications and networked games (Capin et al., 1999; Gutierrez et al., 2003). Due to the large amount of data and the limited bandwidth of communication network, congestion, packet loss, and delay often occur in mocap data transmission. Therefore, mocap data compression, specially lossy compression, is necessary to facilitate storage and transmission.

Thanks to its smooth and coherent nature, mocap data exhibits high degree of temporal and spatial redundancy, making compression possible. To date, many mocap compression algorithms have been proposed (see Section 2). Among these approaches, most are *sequence-based* (e.g., Chattopadhyay et al., 2007; Gu et al., 2009; Tournier et al., 2009; Lin et al., 2011;

Váša and Brunnett, 2014; Hou et al., 2014a, 2015a) in that they process all the frames of a mocap sequence at a time. These methods are able to achieve high compression performance. However, such a good compression performance comes at a price of high latency, i.e., a large number of frames have to be captured and stored before compression, making them more suitable for efficient storage. On the other hand, the *frame-based* (e.g., Kwak and Bajic, 2011) approaches aim at time-critical applications (e.g., interactive applications) due to their no-latency nature. Unfortunately, the existing frame-based methods have poor compressing performance compared with the sequence-based methods, since they cannot explore spatial and temporal correlation well. As none of the sequence- and frame-based methods is perfect, it is natural to consider the *clip-based* (e.g., Arikan, 2006; Liu and McMillan, 2006; Chew et al., 2011) methods which segment mocap data into short clips, providing a trade-off between latency and compression performance.

In this paper, we present two efficient methods for compressing mocap data with low latency. The first method processes the data in a frame-by-frame manner, hereby compressing the data without any inherent latency at all. The second one is clip-based and can achieve higher compression performance while keeping the latency fairly low and controllable. Since mocap data exhibits some unique spatial characteristics, we propose a learned spatial decorrelation transform (LSDT) to explore the spatial redundancy. Taking the data content into account, the LSDT learns an orthogonal matrix via an ℓ_0-norm regularized optimization. Due to its data adapted nature, the proposed LSDT outperforms the commonly used data-independent transforms, such as discrete cosine transform (DCT) and discrete wavelet transform (DWT), in terms of compression performance. We also adopt a predictive coding and temporal DCT for temporal decorrelation in the frame- and clip-based methods, respectively. We observe promising experimental results and demonstrate that our methods can produce higher compression performance at lower computational cost and latency than state-of-the-art.

The rest of this paper is organized as follows: Section 2 comprehensively reviews previous work on mocap data compression. Section 3 gives the proposed frame- and clip-based methods. Section 4 shows the key component of the proposed methods, i.e., the learned spatial decorrelation transform, followed by the experimental results and discussion in Section 5. Finally, Section 6 concludes this paper.

2. Related work

All compression schemes aim at exploiting correlations among the data, so does mocap data compression. In terms of decorrelation techniques, the existing mocap data compression algorithms can be roughly classified into four groups, which are reviewed and analyzed as follows.

2.1. Principal component analysis (PCA)

As a very popular technique, principal component analysis projects the data onto few principal orthogonal bases to convert data into a smaller set of values of linearly uncorrelated data.

Breaking the mocap database into short clips that are approximated by Bézier curves, Arikan (2006) performed clustered PCA to reduce their dimensionality. Liu and McMillan (2006) projected only the keyframes on the PCA bases and interpolated the other frames via spline functions. Motivated by the repeated characteristics of human motions, Lin et al. (2011) projected similar motion clips into PCA space and approximated them by interpolating functions with range-aware adaptive quantization. Observing that distortion to each of the joints causes a different overall distortion, Váša and Brunnett (2014) proposed perception-driven error metric so that important joints have a higher precision than that of joints with small impact. They presented a Lagrange multiplier-based preprocessing for adjusting the joint precision. After Lagrangian equalization, the entire mocap sequence is projected into PCA pose space. Then, PCA is applied to short clips for further reducing the temporal coherence.

Principal geodesic analysis (PGA) is a generalization of PCA for handling the case where the data is sampled from curved manifolds. Tournier et al. (2009) presented a PGA-based method for the poses manifold in the configuration space of a skeleton, leading to a reduced, data-driven pose parameterization. Compression is then obtained by storing only the approximate parameterization along with the end-joints and root-joints trajectories.

Although PCA can decorrelate mocap data very well, its bases are data-dependent and usually difficult to compress. Therefore, one has to explicitly store the orthogonal bases, which reduces the overall compression performance. Furthermore, PCA is usually applied to the whole mocap sequence (e.g., Karni and Gotsman, 2004; Váša and Brunnett, 2014), resulting in a high latency.

2.2. Discrete wavelet and cosine transforms

DCT and DWT are commonly used techniques for converting correlated data into frequency domain, in which energy mainly concentrates on sparse frequencies (or most transform coefficients tend to zero). DCT and DWT have been widely adopted in some video/image coding standards (Wiegand et al., 2003; Sullivan et al., 2012). Moreover, they also have been exploited in the compression of 3D geometric data, e.g., static/dynamic meshes (Gu et al., 2002; Hou et al., 2014b, 2015b) and mocap data (Preda et al., 2007; Chew et al., 2011; Kwak and Bajic, 2011).

Kwak and Bajic (2011) applied 1D DCT to the predictive residuals between consecutive frames for exploiting the spatial coherence. In contrast, Preda et al. (2007) applied 1D DCT/DWT to the residuals of motion compensation along the temporal

dimension. Beaudoin et al. (2007) and Firouzmanesh et al. (2011) adopted 1D DWT to trajectories of degrees of freedom and selected the sparse wavelet coefficients by a perceptual-based metric. Observing that neither 1D DCT nor 1D DWT considers the spatial and temporal correlation simultaneously, Chew et al. (2011) used Fuzzy C-means clustering to represent the mocap clips as 2D arrays, on which 2D DWT was applied.

As pointed out in Hou et al. (2015a), mocap data have some unique features that distinguish them from natural videos/images. For example, applying 1D DCT/DWT to each trajectory produces sparsity in the transform domain, since each trajectory is a smooth spatial curve. However, it does not make sense to apply 1D DCT/DWT to each mocap frame due to the lack of smoothness in the frame (see the analysis in Section 4).

2.3. Mocap data favored transforms

As general-purpose transforms, DWT and DCT are data-*independent* so that one does not need to store the bases. In contrast, data-driven transforms are adaptive to the input data, thus, they can take advantage of their intrinsic structure. However, the adaptiveness comes at a price of storing the basis functions explicitly.

Zhu et al. (2012) proposed an elegant sparse decomposition model for the quaternion space that decomposes human rotational motion into a dictionary part and a weight part. As a result, a linear combination of 3D motion is equivalent to quaternion multiplication and the weight of linear combination is a power operation on quaternion. They showed that the transformed weights are sparse, leading to good compression performance. However, the quaternion space sparse representation is computationally expensive, diminishing its application to long motion sequences. Hou et al. (2014a) represented a motion sequence as a third-order tensor, which exhibits strong correlation within and across its slices. They performed the canonical polyadic (CP) tensor decomposition to explore correlation within and among clips to realize dimensionality reduction. Recently, Hou et al. (2015a) proposed the mocap data tailored transform (MDTT), which partitions the input motion into clips and then computes a set of data-dependent orthogonal bases by minimizing the least square of distortions. Computational results show that MDTT significantly outperforms the existing techniques (e.g., Arikan, 2006; Lin et al., 2011; Zhu et al., 2012; Tournier et al., 2009) in terms of both compression performance and runtime. However, due to the overhead of explicitly storing the orthogonal bases, MDTT is less appealing for the short motion sequence. Note that all of the above-mentioned methods (Zhu et al., 2012; Hou et al., 2014a, 2015a) have very high latency due to their sequence-based nature.

2.4. Indexing-based methods

Chattopadhyay et al. (2007) proposed a smart indexing algorithm for exploiting structural information derived from the human skeleton, where each floating point number is represented as an integer index, based on the statistical distribution of the floating point numbers in a motion matrix. Gu et al. (2009) organized the markers into a hierarchy where each node corresponds to a meaningful part of the human body and coded each body part separately. Then, the motion sequence is represented as a series of motion pattern indices with respect to a predefined dataset including various patterns.

3. Overview

Given a mocap sequence of F frames, we denote its i-th frame by $\mathbf{m}_i^d = [d_1^i \ d_2^i \ \cdots \ d_J^i]^\mathsf{T} \in \mathbb{R}^J$, where J is the number of key points (markers) and $d := \{x, y, z\}$ stands for the d-dimensional coordinate. Then the d-component of the motion sequence is represented by a J-by-F matrix $\mathbf{M}^d = [\mathbf{m}_1^d \ \mathbf{m}_2^d \ \cdots \ \mathbf{m}_F^d] \in \mathbb{R}^{J \times F}$. Each row of $\mathbf{M}^d$ corresponds to the d-trajectory of a key point. We partition $\mathbf{M}^d$ into non-overlapping clips of equal length, denoted by $\widetilde{\mathbf{M}}^d \in \mathbb{R}^{J \times L}$, where L is the clip length.

The primary goal of data compression is to reduce redundancy or correlation in the data. As pointed out in Hou et al. (2015a), a typical mocap sequence exhibits strong spatial correlation due to the highly coordinated and structured nature of key points, and strong local temporal correlation since the object moves smoothly at a relatively small time scale. Therefore, mocap compression aims at eliminating both types of correlation as much as possible. In following sections, we present two low-latency and high-efficiency methods for compressing mocap data.

3.1. Frame-based method

As shown in Fig. 1(a), the frame-based method processes one frame at a time so that there is no inherent latency at all. Let us denote $\mathbf{B}^d$ the basis functions of the learned spatial decorrelation transform (LSDT) (to be presented in Section 4). For the first frame $\mathbf{m}_1$, we use $\mathbf{B}^d$ to remove its spatial correlation, i.e.,

$$\mathbf{c}_1^d = \mathbf{B}^d \mathbf{m}_1^d. \tag{1}$$

Then we adopt a simple predictive coding to the following frames to eliminate the temporal redundancy: the i-th frame is predicted only from the previous reconstructed one

$$\mathbf{r}_i^d = \mathbf{m}_i^d - \widehat{\mathbf{m}}_{i-1}^d, \ (i \geq 2) \tag{2}$$
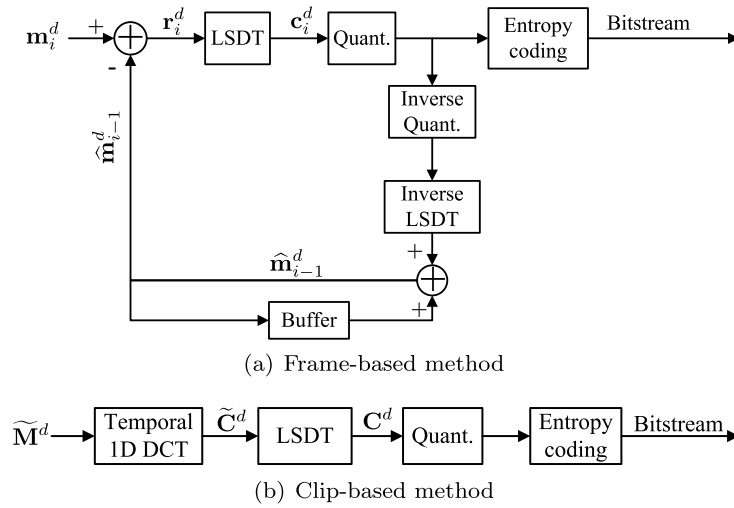
(a) Frame-based method

(b) Clip-based method

Fig. 1. The flowcharts of the proposed frame- and clip-based methods.

where $\widehat{\mathbf{m}}_{i-1}^d$ is the reconstructed $(i-1)$-th frame, which is obtained by inverse quantization and inverse LSDT. Then, applying the spatial decorrelation transform $\mathbf{B}^d$ on the residual vector $\mathbf{r}_i^d$, we obtain

$$\mathbf{c}_i^d = \mathbf{B}^d \mathbf{r}_i^d, \tag{3}$$

where $\mathbf{c}_i^d \in \mathbb{R}^J$ are the transformed coefficients.

Finally, we perform the hard thresholding operation and uniform quantization on $\mathbf{c}_i^d$. We store the following information for reconstruction: (1) the locations and values of nonzero elements, which are further entropy-coded using lossless coding, i.e., Huffman codes; (2) the number of nonzero elements in each coefficient vector, which is encoded using fixed-length encoding.

3.2. Clip-based method

The frame-based scheme has no inherent latency at the price of relatively low compression performance, since it cannot fully exploit the temporal coherence. The clip-based scheme, in contrast, processes L consecutive frames at a time, leading to better temporal decorrelation. With a proper L, the clip-based algorithm is a trade-off between latency and compression performance.

Fig. 1(b) shows the flowchart of the clip-based scheme. Let $\widetilde{\mathbf{M}}^d \in \mathbb{R}^{J \times L}$ be a clip of length L. Each row of $\widetilde{\mathbf{M}}^d$ corresponds to the d dimensional trajectory of a key point, i.e., a spatial curve. Thus, applying the 1D DCT to the rows of $\widetilde{\mathbf{M}}^d$ to explore the temporal correlation (see the analysis in Section 4), we obtain

$$\widetilde{\mathbf{C}}^d = \widetilde{\mathbf{M}}^d \mathbf{U}_t, \tag{4}$$

where $\mathbf{U}_t \in \mathbb{R}^{L \times L}$ is the 1D DCT matrix. We then apply the LSDT to $\widetilde{\mathbf{C}}^d$ to further remove its spatial redundancy,

$$\mathbf{C}^d = \mathbf{B}^d \widetilde{\mathbf{C}}^d. \tag{5}$$

Finally, we adopt the same quantization and entropy coding used in the frame-based method to encode $\mathbf{C}^d$ into bit stream. The sequence can be reconstructed by inverse quantization and inverse transform.

4. Learned spatial decorrelation transform

DCT and DWT decorrelate the data by converting it from spatial domain to frequency domain in a sparse form. They have been widely used for image and video compression (Wiegand et al., 2003; Skodras et al., 2001). DCT is suitable for signals which can be approximately modeled as a first-order Markov process (Markov-I) with the correlation coefficient 1, while DWT is particularly desired to piecewise signals (Wang, 2012). Note that each row of $\mathbf{M}^d$ corresponds to the d-dimensional trajectory of a key point, which can be viewed as Markov-I. Thus, it is reasonable to employ DCT to exploit the coherence within them. However, since the key points are organized in an irregular, tree-like structure (i.e., skeleton graph), the elements of $\mathbf{m}_i^d$ may not be correlated with their neighbors, meaning that columns of $\mathbf{M}^d$ do not follow Markov-I. Also note that the columns of $\mathbf{M}^d$ do not exhibit the piecewise smooth characteristic either. As a result, it does not make sense to apply DCT or DWT for de-correlation among the rows of $\mathbf{M}^d$. We refer readers to Hou et al. (2015a) for quantitative analysis. As pointed out in Safonova et al. (2004), Tan et al. (2013, 2015), mocap data lies in a relatively lower dimensional space,

which are spanned by a set of specific bases. Based on the above analysis, we propose to learn an orthogonal transform to span the subspace of mocap data as much as possible.

Given N training frames $\{\mathbf{m}_i\}_{i=1}^N$, $\mathbf{m}_i \in \mathbb{R}^{J \times 1}$, the learned spatial decorrelation transform (LSDT) aims at finding an orthogonal matrix $\mathbf{B} \in \mathbb{R}^{J \times J}$ so that it can transform each training frame into a sparse vector. We formulate the learning problem as follows:

$$\min_{\substack{\mathbf{B}^d \in \mathbb{R}^{J \times J} \\ \{\mathbf{e}_i^d\} \in \mathbb{R}^J}} \sum_{i=1}^N \left\| \mathbf{B}^d \mathbf{m}_i^d - \mathbf{e}_i^d \right\|_2^2$$

$$\text{subject to} \quad \mathbf{B}^d \mathbf{B}^{d\mathsf{T}} = \mathbf{B}^{d\mathsf{T}} \mathbf{B}^d = \mathbf{I}, \quad \left\| \mathbf{e}_i^d \right\|_0 \leq P, \tag{6}$$

where the ℓ_0-norm $\|\mathbf{e}_i\|_0$ counts the number of non-zero entries in the transform coefficient of the i-th training sample, P is the user-specified parameter controlling the sparsity in $\mathbf{e}_i$, and $\mathbf{I} \in \mathbb{R}^{J \times J}$ is the identity matrix. The orthogonality constraint on $\mathbf{B}^d$ allows us to obtain the inverse LSDT easily. Observe that the optimization problem in Equation (6) is non-convex due to the non-convex constraints. We develop an alternating iterative method, which alternately solves the following two subproblems until convergence.

4.1. The sparse vector subproblem

With fixed $\mathbf{B}^d$, let $\mathbf{g}_i^d \triangleq \mathbf{B}^d \mathbf{m}_i^d$. The sparse vector subproblem is equivalent to the summation of multiple independent univariate minimization problems, in which the i-th one is written as

$$\min_{\{\mathbf{e}_i^d\}} \left\| \mathbf{g}_i^d - \mathbf{e}_i^d \right\|_2^2 \quad \text{subject to} \quad \left\| \mathbf{e}_i^d \right\|_0 \leq P. \tag{7}$$

Obviously, the minimization is achieved only when $\mathbf{e}_i^d$ contains the largest P entries (in magnitude) of $\mathbf{g}_i^d$ which are at the corresponding locations. Therefore, we can compute $\mathbf{e}_i^d$ by setting the $(J - P)$ smallest (in magnitude) entries of $\mathbf{B}^d \mathbf{m}_i^d$ to zero:

$$\mathbf{e}_i^d = \mathcal{T}\left(\mathbf{g}_i^d, J - P \right), \tag{8}$$

where $\mathcal{T}$ is the truncating operation.

4.2. The orthogonal matrix subproblem

Given fixed sparse vectors $\mathbf{e}_i^d$, $i = 1, \ldots, N$, let us denote $\mathbf{E}^d = [\mathbf{e}_1^d, \ldots, \mathbf{e}_N^d]$ the matrix representation. The orthogonal matrix subproblem is

$$\min_{\mathbf{B}^d} \left\| \mathbf{B}^d \mathbf{M}^d - \mathbf{E}^d \right\|_F^2 \quad \text{subject to} \quad \mathbf{B}^d \mathbf{B}^{d\mathsf{T}} = \mathbf{B}^{d\mathsf{T}} \mathbf{B}^d = \mathbf{I}, \tag{9}$$

where $\| \cdot \|_F$ is the Frobenius norm of matrix and $\mathbf{M}^d$ is the matrix representation of all training frames. Observe that

$$\left\| \mathbf{B}^d \mathbf{M}^d - \mathbf{E}^d \right\|_F^2 = \text{Tr}\left(\left(\mathbf{B}^d \mathbf{M}^d - \mathbf{E}^d \right) \left(\mathbf{B}^d \mathbf{M}^d - \mathbf{E}^d \right)^{\mathsf{T}} \right)$$

$$= \text{Tr}\left(\mathbf{M}^d \mathbf{M}^{d\mathsf{T}} \right) - 2\text{Tr}\left(\mathbf{B}^d \mathbf{M}^d \mathbf{E}^{d\mathsf{T}} \right) + \text{Tr}\left(\mathbf{E}^d \mathbf{E}^{d\mathsf{T}} \right),$$

where Tr is the matrix trace.

Ignoring the first and third terms which are constant, the minimization problem in (9) is equivalent to

$$\max_{\mathbf{B}^d} \text{Tr}\left(\mathbf{B}^d \mathbf{M}^d \mathbf{E}^{d\mathsf{T}} \right) \quad \text{subject to} \quad \mathbf{B}^d \mathbf{B}^{d\mathsf{T}} = \mathbf{B}^{d\mathsf{T}} \mathbf{B}^d = \mathbf{I}. \tag{10}$$

Factoring $\mathbf{M}^d \mathbf{E}^{d\mathsf{T}}$ using the singular value decomposition (SVD), we obtain $\mathbf{M}^d \mathbf{E}^{d\mathsf{T}} = \widetilde{\mathbf{U}}^d \mathbf{S}^d \widetilde{\mathbf{V}}^{d\mathsf{T}}$, where $\widetilde{\mathbf{U}}^d$, $\widetilde{\mathbf{V}}^d \in \mathbb{R}^{J \times J}$ are two orthogonal matrices, and $\mathbf{S}^d$ is a diagonal matrix.

Then we can rewrite the objective function as

$$\text{Tr}\left(\mathbf{B}^d \mathbf{M}^d \mathbf{E}^{d\mathsf{T}} \right) = \text{Tr}\left(\mathbf{B}^d \widetilde{\mathbf{U}}^d \mathbf{S}^d \widetilde{\mathbf{V}}^{d\mathsf{T}} \right) = \text{Tr}\left(\widetilde{\mathbf{B}}^d \widetilde{\mathbf{U}}^d \mathbf{S}^d \right),$$

where $\widetilde{\mathbf{B}}^d = \widetilde{\mathbf{V}}^{d\mathsf{T}} \mathbf{B}^d$ is still an orthogonal matrix.

Since $\mathbf{S}^d$ is a diagonal matrix, maximizing (10) is equivalent to maximize the diagonal entries of $\widetilde{\mathbf{B}}^d \widetilde{\mathbf{U}}^d$. With Cauchy–Schwartz inequality, the i-th diagonal entry of $\widetilde{\mathbf{B}}^d \widetilde{\mathbf{U}}^d$ is

Algorithm 1 Computing LSDT bases for mocap data.

Input: training samples $\{\mathbf{m}_i\}_{i=1}^N$, the sparsity parameter P and the maximum number of iterations K
Output: the orthogonal matrix $\mathbf{B}^d$
1: initialize $\mathbf{B}^d$ using an orthogonal matrix (e.g., DCT or DWT bases)
2: **for** $iter \leftarrow 1 : K$ **do**
3: **for** $i \leftarrow 1 : N$ **do**
4:   update $\mathbf{e}_i^d$ using (8)
5: **end for**
6: factor $\mathbf{M}^d\mathbf{E}^{d^\top}$ using SVD
7: update $\mathbf{B}^d$ using (11)
8: **end for**

Table 1
Description of training sequences and test sequences.

Sequence	F	Size (kB)	Description
86_02	10,617	3856.9	walk, squats, run, stretch, jumps,punches, and drinking
56_04	6767	2458.3	fists up, wipe window, grab, walk, throw punches, yawn, stretch, jump
15_05	22,948	8336.5	wash windows, paint, hand signals, dance, dive, twist, boxing
14_08	2625	953.6	jump up to grab
15_04	22,549	8191.6	dance, the twist, boxing
17_08	6179	2244.7	muscular person's walk
17_10	2783	1011	boxing
41_07	7536	2737.7	climb, step over, jump over
49_02	2085	757.4	jump, hop on one foot
56_07	9420	3422.1	yawn, stretch, walk, run, halt
85_12	4499	1634.4	jumps, flips, breakdance
86_05	8340	3029.7	walking, jumping, punching

$$\sum_{j=1}^J \widetilde{\mathbf{B}}_{ij}^d \widetilde{\mathbf{U}}_{ji}^d \le \sqrt{\sum_{j=1}^J \widetilde{\mathbf{B}}_{ij}^{d2} \sum_{j=1}^J \widetilde{\mathbf{U}}_{ji}^{d2}} = 1.$$

The last equation comes from the fact that both $\widetilde{\mathbf{B}}$ and $\widetilde{\mathbf{U}}$ are orthogonal matrices. Therefore, the objective function in (10) is maximized when $\widetilde{\mathbf{B}}^d\widetilde{\mathbf{U}}^d = \mathbf{I}$, leading to

$$\mathbf{B}^d = \widetilde{\mathbf{V}}^d\widetilde{\mathbf{U}}^{d^\top}. \tag{11}$$

Algorithm 1 shows the pseudocode of the LSDT algorithm. In each iteration, the truncating operation (line 4) and matrix multiplication (lines 6 and 7) take $O(J \log J)$ and $O(2NJ^2)$ time, respectively. Singular value decomposition has an $O(J^3)$ time complexity. Putting it all together, the time complexity of Algorithm 1 is $\mathcal{O}(KNJ^2 + KNJ \log J + KJ^3)$. Although there is no theoretical guarantee of the convergence of our algorithm, each subproblem does have an exact solution and we observe that it converges in a few hundred iterations on training datasets (see Section 5.1).

5. Experimental results and discussion

We implement our methods in MATLAB using only 200 lines of codes and evaluate them on the CMU Mocap Database,[1] in which each frame consists of $J = 31$ key points (i.e., joints of the human skeleton) sampled at 120 frames per second (fps). We store each coordinate of the original data as a 32-bit float and hereby represent one key point using 96 bits. Table 1 describes the training and test motion sequences and their lengths.

The compression distortion D is measured by the average Euclidean distance between the original joint location $\mathbf{p}_{i,j} := \{x_{i,j}, y_{i,j}, z_{i,j}\}^\top$ and the reconstructed location $\widehat{\mathbf{p}}_{i,j} := \{\widehat{x}_{i,j}, \widehat{y}_{i,j}, \widehat{z}_{i,j}\}^\top$ (in cm),

$$D = \frac{1}{JF} \sum_{i=1}^J \sum_{j=1}^F \left\| \mathbf{p}_{i,j} - \widehat{\mathbf{p}}_{i,j} \right\|_2. \tag{12}$$

The compression ratio (CR) is the ratio between the original data size and the compressed data size. The compression is determined by the quantization bit, that is, a larger quantization bit induces smaller distortion at a smaller CR.

---

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

(a) 1D DCT bases $\mathbf{U}_t$ (b) LSDT bases $\mathbf{B}^x$ (c) LSDT bases $\mathbf{B}^y$ (d) LSDT bases $\mathbf{B}^z$
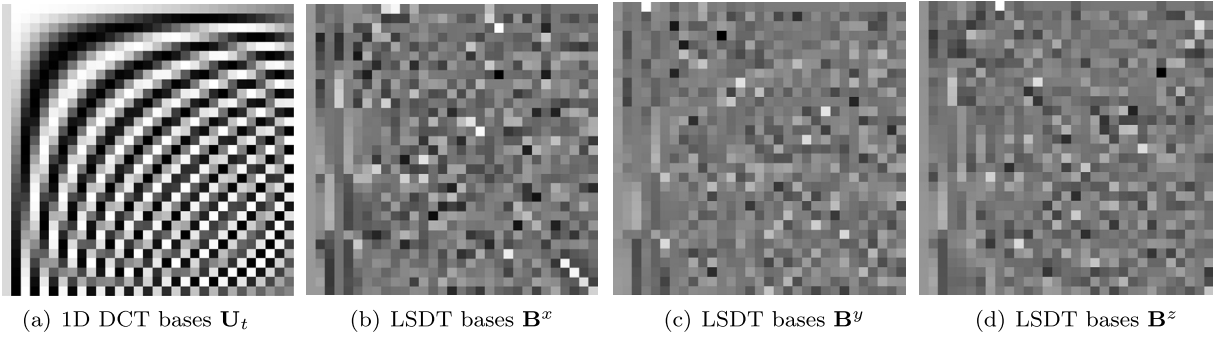
Fig. 2. Visualization of the 1D DCT and LSDT bases, where the greyscale color indicates the normalized function value. In each square matrix, a column corresponds to one basis function and frequencies increase from left to right.
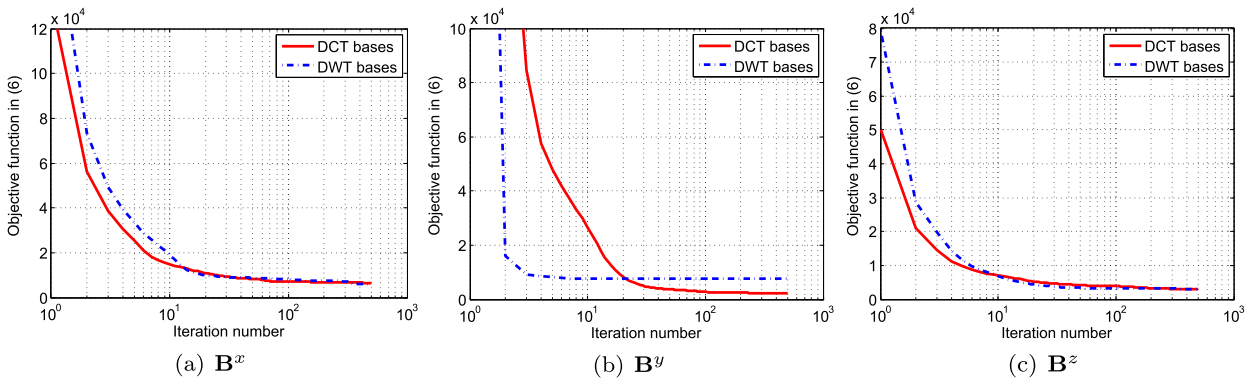

(a) $\mathbf{B}^x$ (b) $\mathbf{B}^y$ (c) $\mathbf{B}^z$

Fig. 3. Convergence plots of Algorithm 1 with two different initializations. # Training frames $N = 10{,}617$; sparsity parameter $P = 8$. (a), (b), and (c) Correspond to x, y, and z-coordinates, respectively.

5.1. Training the LSDT bases

We take sequences "86_02" "56_04", and "15_05" as the training datasets, which consist of various types of human motion. It is worth noting that more training frames can generate better performance, but the computational cost also increases. Thus, it is a tradeoff between quality and efficiency.

The LSDT bases training algorithm (cf. Algorithm 1) is an iterative algorithm. We evaluate the convergence rate of the training algorithm on two types of initializations, 1D DCT bases and 1D DWT bases realized by the 3-level "Haar" wavelet. As Fig. 3 shows, the objective function converges to almost the same value after a few hundred iterations, meaning that the output of Algorithm 1 is intrinsic, which does not depend on initialization. Fig. 2 also visualizes the bases of 1D DCT and LSDT to show the difference between them.

The parameter P, specifying the sparsity of transform coefficients during the learning procedure, directly affects the structure of the learned orthogonal matrix $\mathbf{B}^d$, which in turn controls the compression performance. In the training process, we set P to four different values: 2, 5, 8, and 11. Then, the learned orthogonal matrices under different P are tested in the frame- and clip-based methods, respectively. For both schemes, four randomly chosen sequences with various motion characteristics and lengths are compressed, and the results are shown in Fig. 4, where we can see that the best compression performance is achieved when the value of P is equal to 8.

5.2. Evaluating the spatial decorrelation transforms

We compare the performance of several spatial decorrelation transforms, including LSDT, spatial DCT, and spatial DWT. We apply each transform to the x, y and z components of each frame separately, and examine the relationship between the percentage of nonzero transform coefficients and the distortion. As Fig. 5 shows, given the same number of nonzero transformed coefficients, the distortions produced by LSDT are consistently much smaller than those of DCT and DWT, meaning that LSDT concentrates energy (or spatially decorrelated mocap data) better than DCT and DWT.
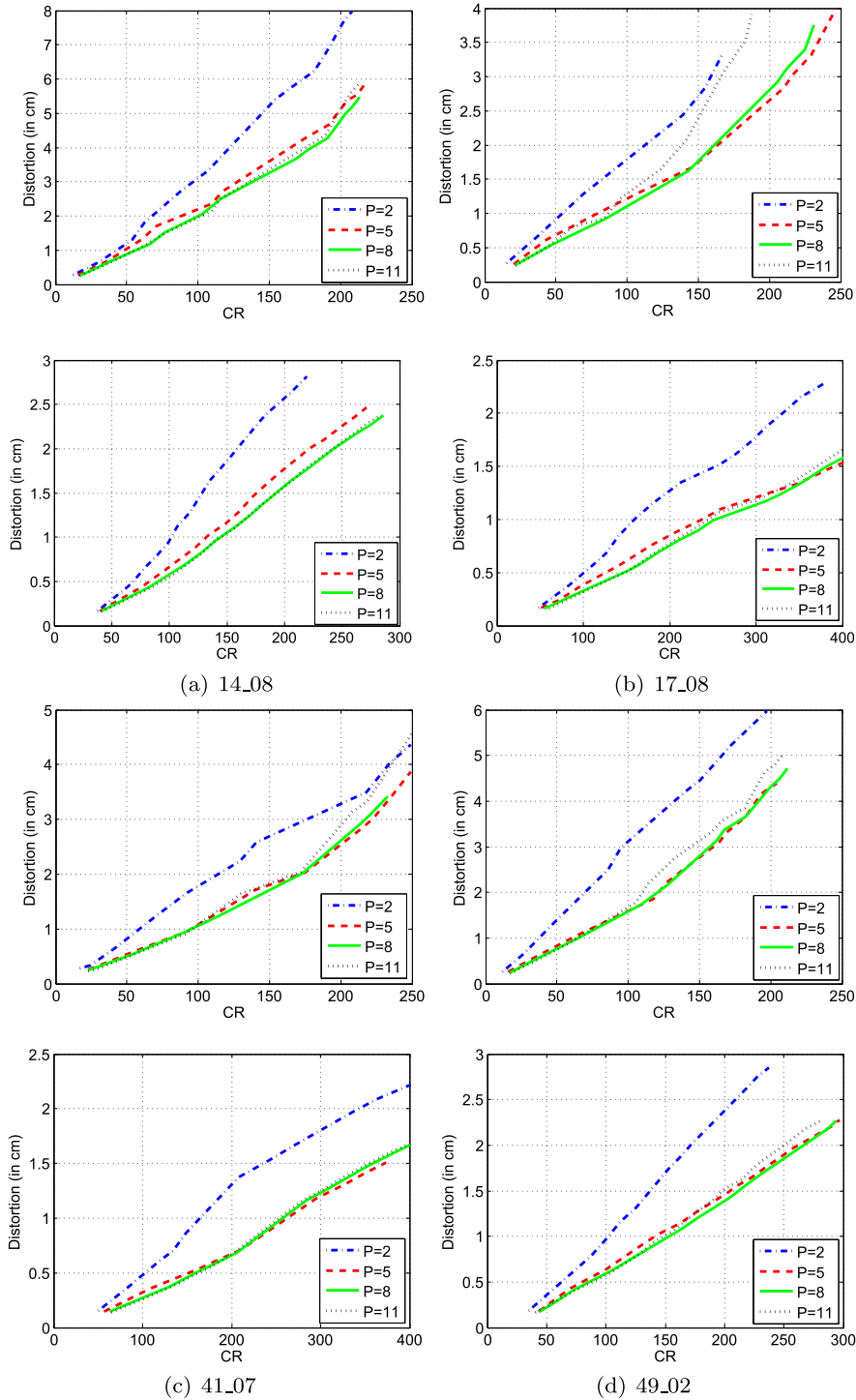
Fig. 4. The impact of the sparsity parameter P on the overall compression performance. The top and bottom rows correspond to the frame- and clip-based ($L = 240$) schemes, respectively. $\mathbf{B}^d$ is initialized using the DCT bases.

5.3. Compression performance

Fig. 6 shows the CR-distortion (CR-D) curves of the frame-based scheme. As Section 5.2 shows, our data-adapted LSDT is superior to the data-independent 1D DCT for spatial decorrelation. Therefore, it is not surprising that our frame-based scheme significantly outperforms the 1D DCT-based method (Kwak and Bajic, 2011) in terms of compression performance.
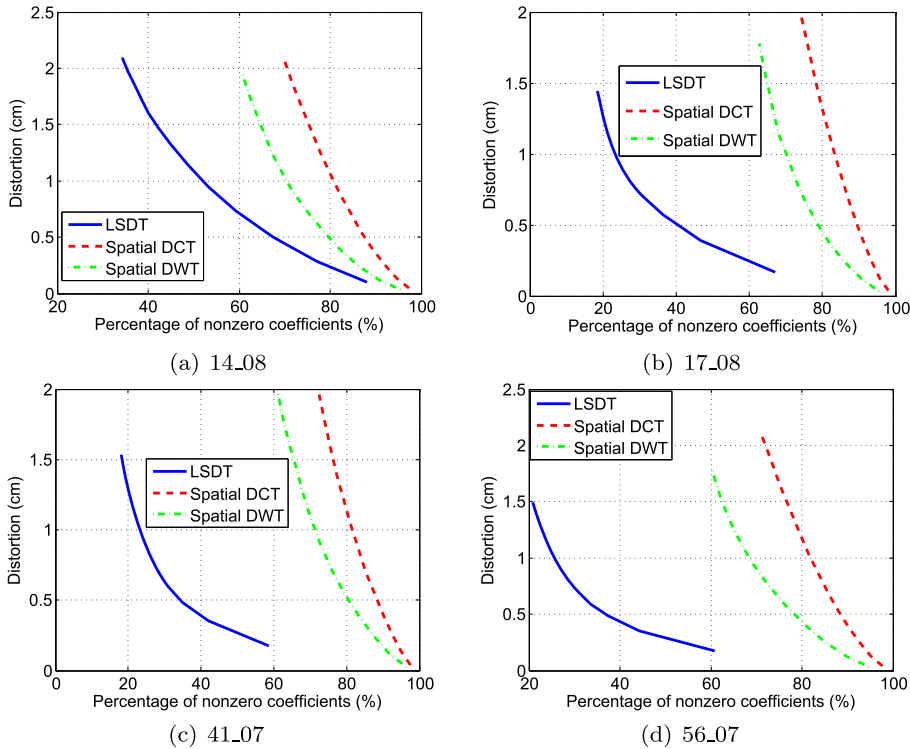
Fig. 5. Evaluating the performance of spatial decorrelation of the proposed LSDT. The horizontal axis shows the percentage of nonzero transformed coefficients. LSDT performs the best among the three SDTs.

We observe that with a relatively high CR, our frame-based scheme can reduce up to 70% distortion of Kwak and Bajic (2011).

Fig. 7 shows the CR-D curves of the clip-based scheme, from which we observe the following:

1. As expected, the clip-based scheme has much better compression performance than the frame-based scheme, since it can exploit the temporal coherence better. At the same time, users can easily control the latency for the clip-based scheme. Taking the CMU mocap data which are sampled at 120 fps as example, the clip length $L = 120$ (resp. 240) means 1 second (resp. 2 seconds) latency.

2. The compression performance of the proposed clip-based scheme can be improved by increasing the clip length (or latency). More specifically, when L ranges from 60 to 120, the trajectories in a clip still remain smooth and have small variation (due to the small duration), causing the DCT coefficients to be distributed at similar locations, which can then be encoded using a similar number of bits. Since the number of clips in the sequence decreases, the total number of bits to encode one sequence (i.e., the sum of bits for all clips in one sequence) is significantly reduced, leading to higher compression performance. However, the improvement is little when the value of L increases from 120 to 240. The reason is that the joint trajectories change more significantly. As a result, the DCT coefficients are spread out, requiring more bits for encoding. Although the number of clips decreases, the total number of bits for one sequence only drops slightly.

5.4. Comparison

Table 2 qualitatively compares our methods with the existing works in terms of latency, computational cost, implementation, compression performance, and the number of parameters used in the **encoding** process. Note that **all** methods have a quantization parameter to specify the number of bits used to quantize a coefficient. We do not include this quantization parameter in Table 2, since it is a fixed parameter according to bandwidth. Also note that the sparsity parameter P in our method appears only in the **training** stage.

In this subsection, we compare our clip-based scheme with only two works, namely PCA-Rate Distortion Optimization (PCA-RDO) method (Váša and Brunnett, 2014), and the equal segmentation case of Mocap Data Tailored Transform (MDTT) method (Hou et al., 2015a), which represent the state-of-the-art. See Váša and Brunnett (2014) and Hou et al. (2015a) for detailed performance evaluation on earlier works (Arikan, 2006; Lin et al., 2011; Tournier et al., 2009; Gu et al., 2009; Zhu et al., 2012).
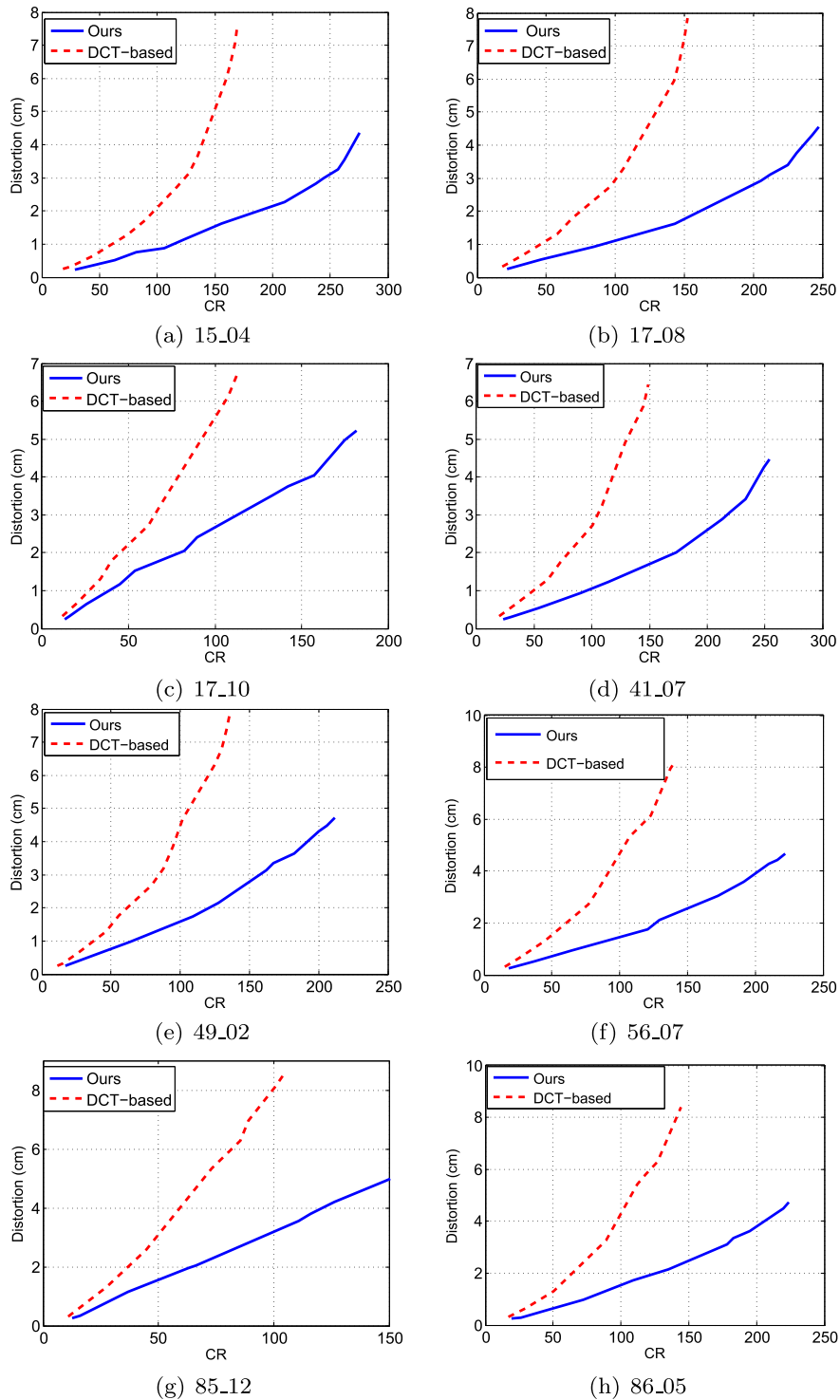
Fig. 6. Comparison of compression performance of frame-based methods.

5.4.1. Comparison with the MDTT method

Both our clip-based algorithm and the MDTT method (Hou et al., 2015a) apply temporal DCT to each trajectory for *temporal* decorrelation. The two methods differ fundamentally in spatial decorrelation. For each mocap sequence, the MDTT method segments the motion sequence into short clips, and compute a set of orthogonal basis functions tailored for all clips together, resulting in better decorrelation at the price of a large latency and overhead for storing the data-dependent basis
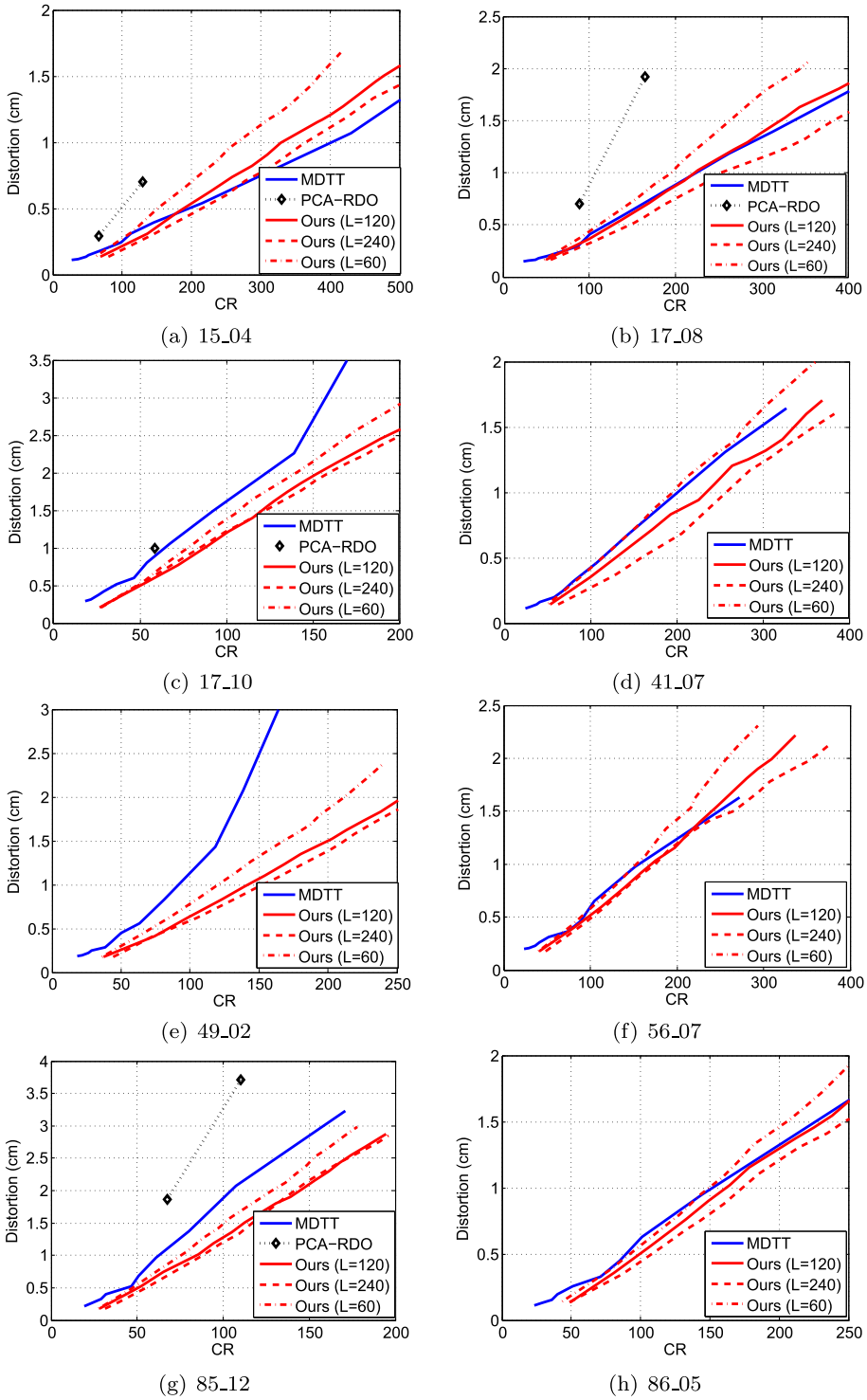
Fig. 7. Compression performance of our clip-based schemes and the state-of-the-art methods, such as the PCA–RDO method (Váša and Brunnett, 2014) and the MDTT method (Hou et al., 2015a). For MDTT, we adopt equal segmentation with $L = 240$ and follow (Hou et al., 2015a) to set the other parameters. The results of PCA–RDO were taken from (Váša and Brunnett, 2014).

Table 2

Qualitative comparison of various mocap compression methods. The latency is measured in number of frames. $\#p$: the number of parameters used in the encoding process; F_s: the number of frames in a mocap sequence; F_c: the numbers of frames in a short clip, and $F_s \gg F_c$. Note that the quantization parameter is not included in $\#p$ for all methods.

Category	Method	Latency	$\#p$	Computational	Implementation cost	Compression performance
PCA-based	Váša and Brunnett (2014)	F_s	5	high	fair	high
	Lin et al. (2011)	F_s	3	fair	difficult	medium
	Arikan (2006)	F_c	3	fair	fair	low
	Liu and McMillan (2006)	F_c	3	fair	fair	low
	Tournier et al. (2009)	F_s	2	high	fair	medium
DCT/ DWT-based	Kwak and Bajic (2011)	0	0	low	easy	low
	Chew et al. (2011)	F_c	2	fair	easy	medium
	Firouzmanesh et al. (2011)	F_c	3	low	easy	low
Mocap data favored transform	Zhu et al. (2012)	F_s	3	high	difficult	medium
	Hou et al. (2015a)	F_s	2	low	easy	high
	Hou et al. (2014a)	F_s	2	fair	fair	medium
	Our frame-based method	**0**	**0**	**low**	**easy**	**high**
	Our clip-based method	$\boldsymbol{F_c}$	**1**	**low**	**easy**	**high**
Indexing-based	Chattopadhyay et al. (2007)	F_s	3	fair	fair	low
	Gu et al. (2009)	F_s	4	fair	fair	low

functions. Within our clip-based method, the LSDT bases are adapted to all mocap data, therefore, there is no need to store the bases for each sequence.

The MDTT method adopts low-rank approximation, which is a linear approximation, to reduce the dimension of transformed coefficients. In contrast, the LSDT makes the transform coefficients sparse by quantization, which is a nonlinear approximation and more flexible. It has been pointed out in Donoho et al. (1998), Cohen et al. (2002) that the nonlinear approximation outperforms the linear approximation in data compression.

From the CR-D curves in Fig. 7, we observe the MDTT (Hou et al., 2015a) has better performance than our scheme for long motion sequences (e.g., 15_04 and 56_07), where the overhead of storing MDTT bases (compared with the transformed coefficients) is very small so that it can be ignored. However, for short sequences (e.g., 17_10 and 49_02), the space usage for storing the basis functions in the MDTT is comparable to that of the transformed coefficients, leading to a large overhead. As a result, its compression performance is not as good as ours. For remaining sequences, the MDTT method is comparable to ours.

Our clip-based method and the MDTT method have similar runtime performance, which can process more than 10,000 frames per second on an Intel Core i7-3770 CPU (3.40 GHz).

In summary, both methods have merits. The mocap tailored transform is suitable for long motion sequences in the applications where large latency is tolerated, while our methods work for both short and long sequences and are desired for time-critical applications such as streaming.

5.4.2. Comparison with the PCA-RDO method

The PCA-RDO method (Váša and Brunnett, 2014) is a PCA-based approach, which adopts PCA twice. In the first round, it applies PCA to the entire motion sequence to obtain reduced orthogonal basis of pose space. This PCA, called posed space PCA, is to explore the spatial correlation. Then, applying PCA to clips, it obtains orthogonal basis for joint trajectories. The second PCA, called temporal PCA, is for temporal decorrelation. With two rounds of PCA, the data dimension is reduced significantly. Váša and Brunnett (2014) also proposed a general preprocessing step based on Lagrange multipliers, which allows the user to optimize with respect to various error metrics.

Our clip-based method and the PCA-RDO method differ in several aspects: First, the PCA-RDO method is sequence-based, thus, it has large latency, whereas ours is clip-based and has low latency. Second, it is known that compression of the PCA's orthogonal basis is difficult, although their method adopts an advanced predictive coding (Váša and Skala Cobra, 2009). As Figs. 7(a), (b), (c), (g) show, our clip-based scheme consistently outperforms the PCA-RDO method (Váša and Brunnett, 2014) in terms of compression performance. Third, similar to the MDTT method, the PCA-RDO method is also low-rank approximation-based. So, it is not as flexible as ours. Fourth, the PCA-RDO algorithm has high computational cost and we observe that the speed of our clip-based method is 3 to 4 times faster than theirs. Last but not least, tuning the parameters of the PCA-RDO method is tedious and non-intuitive. In contrast, within our clip-based method, the user only needs to specify the clip length L, which directly controls the latency.

Finally, Figs. 8 and 9 show some visual results of our methods, the DCT-based, and the MDTT to further demonstrate the advantage of our methods.

5.5. Discussion

We formulate the LSDT problem as a least square with *orthogonal* constraint. In fact, a non-orthogonal matrix $\mathbf{B}^d$ may produce even better compression performance. However, one has to employ other constraints (e.g., using the determinant

(a) 56_07, CR=105

(b) 85_12, CR=90
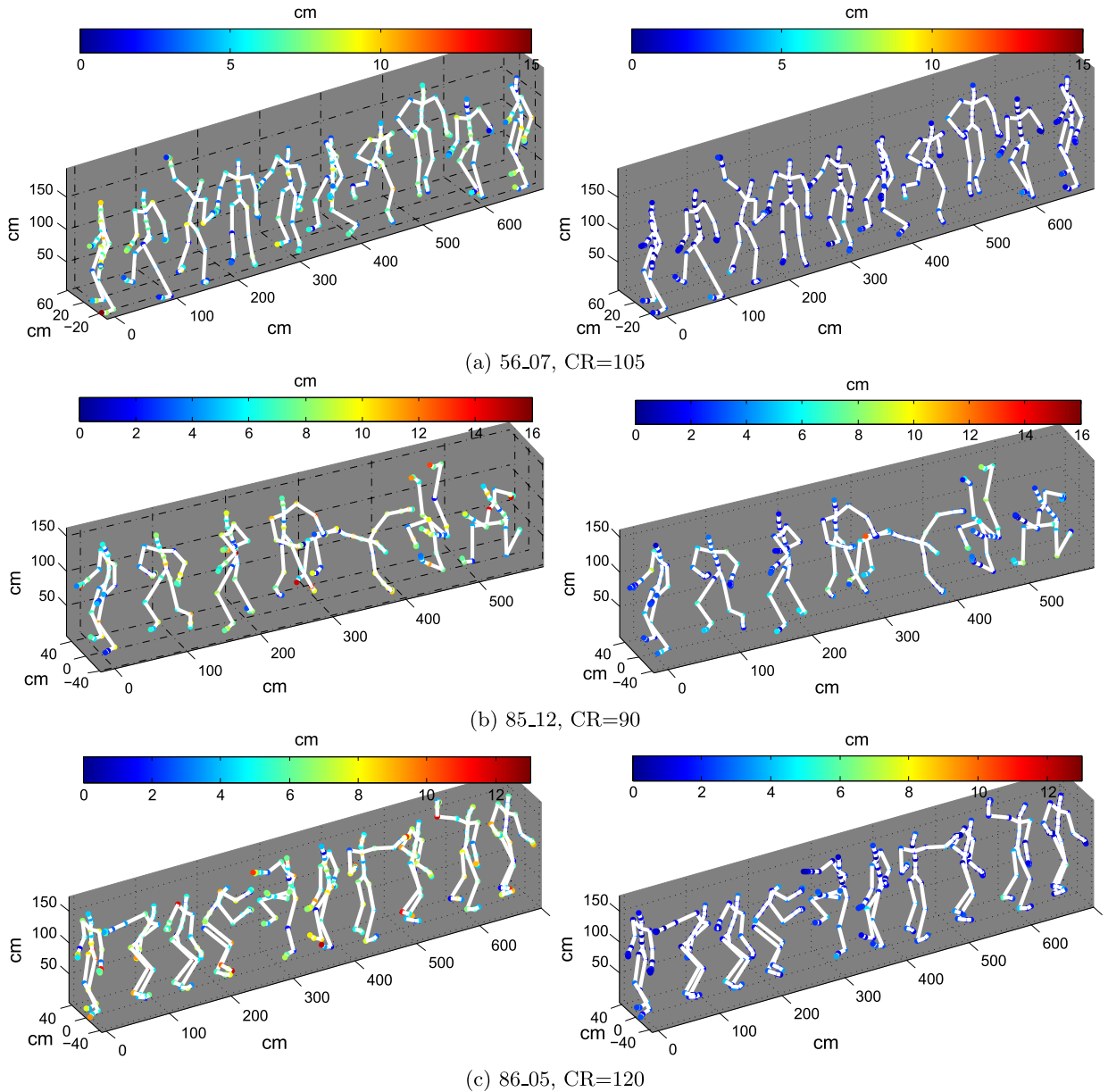

(c) 86_05, CR=120

Fig. 8. Visual results comparison of frame-based schemes. The distortions are colored as heat map, and the frames are uniformly extracted from the sequences. Left: the DCT-based method in Kwak and Bajic (2011); Right: our frame-based scheme.

of $\mathbf{B}^d$ and Frobenius norm of $\mathbf{B}^d$) to ensure the learned matrix invertible (i.e., ensure existence of the inverse transform) and a small condition number. Correspondingly, the optimization problem becomes complicated and it is difficult to solve.

6. Conclusion

We presented frame- and clip-based methods for compressing mocap data with low latency. Taking advantage of the unique spatial characteristics, we proposed learned spatial decorrelation transform to effectively reduce the spatial redundancy in mocap data. Due to its data adaptive nature, LSDT outperforms the commonly used *data-independent* transforms, such as discrete cosine transform and discrete wavelet transform, in terms of the decorrelation performance. Experimental results show that the proposed methods can produce higher compression ratios at a lower computational cost and latency than the state-of-the-art methods.

In our current implementation, we compress 3D *position-based* mocap data defined on a skeleton graph. However, it is straightforward to apply our methods to other types of mocap data, such as facial expressions, hand gestures and motion
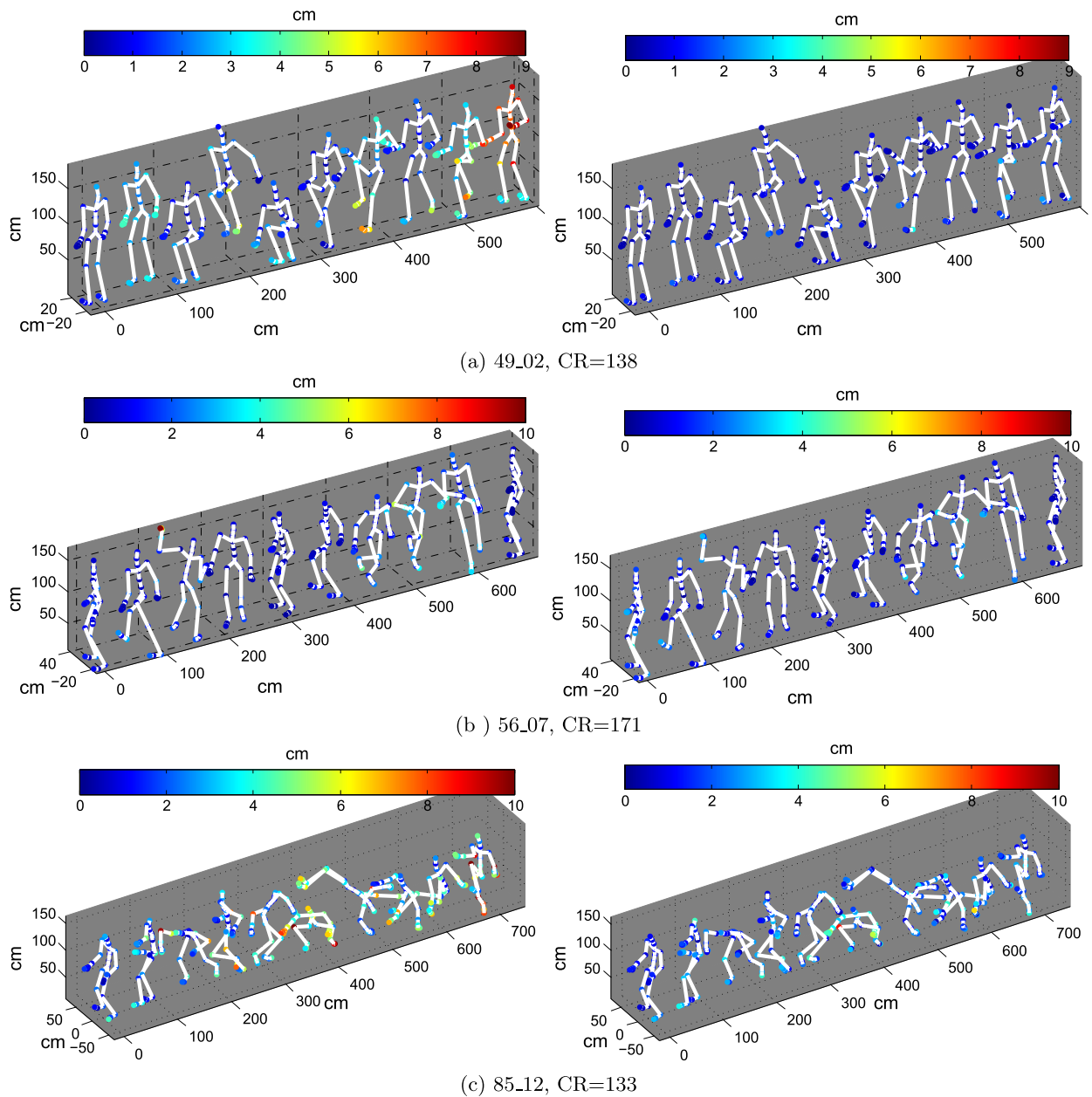
(a) 49_02, CR=138

(b) 56_07, CR=171

(c) 85_12, CR=133

Fig. 9. Visual results comparison of our clip-based scheme and the MDTT method (Hou et al., 2015a). The joint distortions are colored in heat map, and the frames are uniformly extracted from the test sequences. Left: MDTT; Right: our clip-based scheme.

of human bodies. In the future, we will extend our methods to compress mocap data represented by Euler angles. Due to the nonlinear nature of angles, the hierarchical structure may produce significant accumulation errors in the compressed data (Arikan, 2006; Chew et al., 2011). We will seek effective data-driven techniques to tackle this challenge.

Acknowledgements

References

Arikan, O., 2006. Compression of motion capture databases. ACM Trans. Graph. 25 (3), 890–897.

Beaudoin, P., Poulin, P., van de Panne, M., 2007. Adapting wavelet compression to human motion capture clips. In: Proceedings of Graphics Interface, pp. 313–318.

Capin, T., Pandžić, I., Magnenat-Thalmann, N., Thalmann, D., 1999. Avatars in Networked Virtual Environments. John Wiley & Sons.

Chattopadhyay, S., Bhandarkar, S., Li, K., 2007. Human motion capture data compression by model-based indexing: a power aware approach. IEEE Trans. Vis. Comput. Graph. 13 (1), 5–14.

Chew, B.-S., Chau, L.-P., Yap, K.-H., 2011. A fuzzy clustering algorithm for virtual character animation representation. IEEE Trans. Multimed. 13 (1), 40–49. http://dx.doi.org/10.1109/TMM.2010.2082512.

Cohen, A., Daubechies, I., Guleryuz, O.G., Orchard, M.T., 2002. On the importance of combining wavelet-based nonlinear approximation with coding strategies. IEEE Trans. Inf. Theory 48 (7), 1895–1921.

Donoho, D.L., Vetterli, M., DeVore, R.A., Daubechies, I., 1998. Data compression and harmonic analysis. IEEE Trans. Inf. Theory 44 (6), 2435–2476.

Firouzmanesh, A., Cheng, I., Basu, A., 2011. Perceptually guided fast compression of 3-D motion capture data. IEEE Trans. Multimed. 13 (4), 829–834. http://dx.doi.org/10.1109/TMM.2011.2129497.

Gu, X., Gortler, S.J., Hoppe, H., 2002. Geometry images. ACM Trans. Graph. 21 (3), 355–361.

Gu, Q., Peng, J., Deng, Z., 2009. Compression of human motion capture data using motion pattern indexing. Comput. Graph. Forum 28 (1), 1–12.

Gutierrez, M., Vexo, F., Thalmann, D., 2003. Controlling virtual humans using PDAs. In: Proceedings of the 9th International Conference on Multi-Media Modeling, pp. 150–166.

Hou, J., Chau, L.-P., Magnenat-Thalmann, N., He, Y., 2014a. Scalable and compact representation for motion capture data using tensor decomposition. IEEE Signal Process. Lett. 21 (3), 255–259. http://dx.doi.org/10.1109/LSP.2014.2299284.

Hou, J., Chau, L.-P., Zhang, M., Magnenat-Thalmann, N., He, Y., 2014b. A highly efficient compression framework for time-varying 3-D facial expressions. IEEE Trans. Circuits Syst. Video Technol. 24 (9), 1541–1553. http://dx.doi.org/10.1109/TCSVT.2014.2313890.

Hou, J., Chau, L.-P., Magnenat-Thalmann, N., He, Y., 2015a. Human motion capture data tailored transform coding. IEEE Trans. Vis. Comput. Graph. 21 (7), 848–859.

Hou, J., Chau, L., Magnenat-Thalmann, N., He, Y., 2015b. Compressing 3-D human motions via keyframe-based geometry videos. IEEE Trans. Circuits Syst. Video Technol. 25 (1), 51–62. http://dx.doi.org/10.1109/TCSVT.2014.2329376.

Karni, Z., Gotsman, C., 2004. Compression of soft-body animation sequences. Comput. Graph. 28 (1), 25–34.

Kwak, C.-H., Bajic, I.V., 2011. Hybrid low-delay compression of motion capture data. In: Proceedings of IEEE International Conference on Multimedia and Expo. ICME, pp. 1–6.

Lin, I.-C., Peng, J.-Y., Lin, C.-C., Tsai, M.-H., 2011. Adaptive motion data representation with repeated motion analysis. IEEE Trans. Vis. Comput. Graph. 17 (4), 527–538. http://dx.doi.org/10.1109/TVCG.2010.87.

Liu, G., McMillan, L., 2006. Segment-based human motion compression. In: Proceedings of the ACM SIGGRAPH/Eurographics SCA, pp. 127–135.

Preda, M., Jovanova, B., Arsov, I., Prêteux, F., 2007. Optimized MPEG-4 animation encoder for motion capture data. In: Proceedings of the International Conference on 3D Web Technology, pp. 181–190.

Safonova, A., Hodgins, J.K., Pollard, N.S., 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. ACM Trans. Graph. 3, 514–521.

Skodras, A., Christopoulos, C., Ebrahimi, T., 2001. The JPEG 2000 still image compression standard. IEEE Signal Process. Mag. 18 (5), 36–58.

Sullivan, G., Ohm, J., Han, W.-J., Wiegand, T., 2012. Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. 22 (12), 1649–1668. http://dx.doi.org/10.1109/TCSVT.2012.2221191.

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

Tan, C.-H., Hou, J., Chau, L.-P., 2015. Motion capture data recovery using skeleton constrained singular value thresholding. Vis. Comput. 31 (11), 1521–1532.

Tournier, M., Wu, X., Courty, N., Arnaud, E., Reveret, L., 2009. Motion compression using principal geodesics analysis. Comput. Graph. Forum 28 (2), 355–364.

Váša, L., Brunnett, G., 2014. Rate-distortion optimized compression of motion capture data. Comput. Graph. Forum 33 (2), 283–292.

Váša, L., Skala Cobra, V., 2009. Compression of the basis for PCA represented animations. Comput. Graph. Forum 28 (6), 1529–1540.

Wang, R., 2012. Introduction to Orthogonal Transforms: With Applications in Data Processing and Analysis. Cambridge University Press.

Wiegand, T., Sullivan, G., Bjontegaard, G., Luthra, A., 2003. Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol. 13 (7), 560–576. http://dx.doi.org/10.1109/TCSVT.2003.815165.

Zhu, M., Sun, H., Deng, Z., 2012. Quaternion space sparse decomposition for motion compression and retrieval. In: Proceedings of the ACM SIGGRAPH/Eurographics SCA, pp. 183–192.