Human Motion Capture Data Compression based on Learned Spatial Transform

Junhui Hou, Lap-Pui Chau, Nadia Magnenat-Thalmann, and Ying He

Abstract—Due to the growing needs of human motion capture (mocap) in movie, video games, sports, etc., it is highly desired to compress mocap data for efficient storage and transmission. This paper presents two efficient frameworks for compressing human mocap data with low latency. The first framework processes the data in a frame-by-frame manner so that it is ideal for mocap data streaming and time critical applications. The second one is clip-based and provides a flexible trade-off between latency and compression performance. Since mocap data exhibits some unique spatial characteristics, we propose a very effective transform, namely learned orthogonal transform (LOT), for reducing the spatial redundancy. The LOT problem is formulated as minimizing square error regularized by orthogonality and sparsity and solved via alternating iteration. We also adopt a predictive coding and temporal DCT for temporal decorrelation in the frame- and clip-based frameworks, respectively. Experimental results show that the proposed frameworks can produce higher compression performance at lower computational cost and latency than the state-of-the-art methods.

Index Terms—Motion capture, data compression, transform coding, spatial decorrelation, low latency

1 INTRODUCTION

HUMAN motion capture (mocap) is the process of digitally recording human movements by tracking passive or active markers on the subject’s body segments. As a highly successful technique, human mocap data has been widely used to animate virtual human-like characters in distributed virtual reality applications and networked games [1], [2]. 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 the smooth and coherent nature, human 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., [3], [4], [5], [6], [7], [8], [9]) in that all frames of one mocap sequence are simultaneously processed. These methods sometimes can 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 performing compression, making them more suitable for efficient storage. On the other hand, the frame-based (e.g., [10]) approaches aim at time critical applications (e.g., interactive applications) thanks 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 both 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., [11], [12], [13]) schemes which segment mocap data into short clips, providing a trade-off between latency and compression performance.

In this paper, we present two efficient frameworks for compressing human mocap data with low latency. The first framework processes the data frame by frame with no latency. 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 very effective transform, namely learned orthogonal transform (LOT), to explore the spatial redundancy. The LOT learns an orthogonal matrix via $\ell_0$-norm regularized optimization, which takes the data content into account. Thanks to its data adapted nature, the proposed transform outperforms the commonly used data-independent transforms, such as discrete cosine transform (DCT) and discrete wavelet transform (DWT), in terms of spatial decorrelation. We also adopt a predictive coding and temporal DCT for temporal decorrelation in the frame- and clip-based frameworks, respectively. We observe promising experimental results and demonstrate that our methods can produce higher compression performance in lower computational cost and latency than the state-of-the-art methods.

The rest of this paper is organized as follows: Section 2 comprehensively reviews previous work on human

This work carried out at BeingThere Center was supported by the Singapore National Research Foundation, under its International Research Centre at the Singapore Funding Initiative, and administered by the IDM Programme Office.

J. Hou and L.-P. Chau are with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, 639798. Email: houj0001@ntu.edu.sg, elpchau@ntu.edu.sg

N. Magnenat-Thalmann is with the Institute for Media Innovation, Nanyang Technological University, Singapore, 639798. Email: nadathalmann@ntu.edu.sg

Y. He is with the School of Computer Engineering, Nanyang Technological University, Singapore, 639798. Email:yhe@ntu.edu.sg

arXiv:1506.08898v2 [cs.MM]  10 Jul 2015
mocap data compression. Section 3 gives the proposed frame- and clip-based frameworks. Section 4 shows the key component of the proposed frameworks, i.e., an effective spatial decorrelation transform, namely learned orthogonal 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 the correlations among the data, so does mocap data compression. In terms of the 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-based Methods

As a very popular technique, principal component analysis (PCA) 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 [11] performed clustered PCA to reduce their dimensionality. Liu and McMillan [12] 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. [6] 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 [7] 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. [5] 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 compromises the overall compression performance. Besides, PCA is usually applied to the whole sequence (e.g., [14], [15], [7]), resulting in a high latency.

2.2 Discrete Wavelet/Cosine Transform-based Methods

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 [16], [17]. Moreover, they also have been exploited in the compression of 3D geometric data, e.g., static/dynamic meshes [18], [19], [20] and mocap data [21], [13], [10].

Kwak and Bajic [10] applied 1D DCT to the predictive residuals between consecutive frames for exploiting the spatial coherence. In contrast, Preda et al. [21] applied 1D DCT/DWT to the residuals of motion compensation along the temporal dimension. Beaudoin et al. [22] and Firouzianesh et al. [23] adopted 1D DWT to the joints trajectories and selected the sparse wavelet coefficients by perceptual-based metric. Observing that neither 1D DCT nor 1D DWT considers the spatial and temporal correlation simultaneously, Chew et al. [13] used Fuzzy C-means clustering to represent the mocap clips as 2D arrays, on which 2D DWT was applied.

As pointed out in [8], mocap data have some unique features that distinguish them from natural videos/images. For example, applying 1D DCT/DWT to each joint 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 lack of smoothness in the frame.

2.3 Mocap Data Favored Transform-based Methods

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. [24] 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. [9] 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. [8] 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., [11], [6], [24], [5]) in terms of both compression performance and runtime. However, due to the overhead of explicitly storing the orthogonal bases, MDTT is less appealing to the short motion sequence. Note that all of the above-mentioned methods [24], [9], [8] have very high latency due to their sequence-based nature.

2.4 Indexing-based Methods

Chattopadhyay et al. [3] 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. [4] 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

Note that the mocap data compressed in this paper refers to 3D positions (or coordinates) of a set of key points at sampling instants, such as joints of human skeleton and markers placed on subjects during capturing [1]. Given a mocap sequence of \( F \) frames, we denote the \( i \)-th frame by \( m_i^d = [d_1, d_2, \ldots, d_J]^T \in \mathbb{R}^J \), where \( J \) is the number of key points 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 \( M^d = [m_1^d, m_2^d, \ldots, m_F^d] \in \mathbb{R}^{J \times F} \). Each row of \( M^d \) corresponds to the \( d \)-trajectory of a key point. We partition \( M^d \) into non-overlapping clips of equal length, denoted by \( \hat{M}^d \in \mathbb{R}^{J \times L} \), where \( L \) is the clip length.

The primary goal of data compression is to reduce the redundancy or correlation in the data. As pointed out in [8], a typical mocap sequence exhibits strong spatial correlation due to the highly coordinated and structured nature of human motions, and strong local temporal correlation since the human body 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 frameworks for compressing human mocap data.

3.1 Frame-based Framework

As shown in Figure 1(a), the frame-based framework processes one frame at a time so that there is no latency. Let us denote \( B^d \) the learned orthogonal transform for spatial decorrelation (the details will be presented in Section 4). For the first frame \( m_1 \), we use \( B^d \) to remove its spatial correlation, i.e.,

\[ c_1^d = B^d m_1^d. \]

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

\[ r_i^d = m_i^d - \hat{m}_{i-1}^d, \quad (i \geq 2) \]

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

\[ c_i^d = B^d r_i^d, \]

where \( c_i^d \in \mathbb{R}^J \) are the transformed coefficients.

Finally, we quantize each element of \( c_i^d \) using \( q \)-bits. 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.

One can also adopt some advanced technologies in video compression [17] to further improve the performance, such as multiple reference frames-based predictive coding and context adaptive binary arithmetic entropy coding.
3.2 Clip-based Framework

The frame-based scheme has no 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, resulting in better temporal decorrelation. With a proper \( L \), the clip-based algorithm is a trade-off between latency and compression performance.

Figure 1(b) shows the flowchart of the clip-based scheme. Let \( \tilde{M} \in \mathbb{R}^{J \times L} \) be a clip of length \( L \). Each row of \( \tilde{M} \) corresponds to the \( d \) dimensional trajectory of a key point (i.e., a spatial curve), which is a signal defined on the 1D chain graph. According to [25], 1D DCT is optimal for reducing the correlation for such signal under the Gaussian Markov random field assumption. Thus, applying the 1D DCT to the rows of \( \tilde{M} \) to explore the temporal correlation, we obtain

\[
\tilde{C}^d = \tilde{M}^d U_t, \quad (4)
\]

where \( U_t \in \mathbb{R}^{L \times L} \) is the 1D DCT matrix. We then apply the LOT to \( \tilde{C}^d \) to further remove its spatial redundancy,

\[
C^d = B^d \tilde{C}^d. \quad (5)
\]

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

4 PROPOSED SPATIAL DECORRELATION TRANSFORM

4.1 Motivation

Although mocap data is conveniently represented in a 2D array \( M^d \), like images and videos, the difference between the two types of data is fundamental: natural images and videos are homogeneous in that they are smooth in both the horizontal and vertical dimensions, whereas mocap data is heterogeneous, since their horizontal dimension (i.e., temporal) is smooth but the vertical dimension (i.e., spatial) is not (see [3] for a detailed discussion). As a result, naively borrowing the commonly used image/video compression schemes, such as DCT and DWT, does not make sense to mocap data. Besides, compared to images/videos, mocap data lies in relatively lower dimensional space [26]. Based on the above analysis, we propose to learn an orthogonal transform to effectively explore the spatial correlation.

4.2 Learned Orthogonal Transform

Given \( N \) training frames \( \{m_i\}_{i=1}^N \), \( m_i \in \mathbb{R}^{J \times 1} \), learned orthogonal transform (LOT) aims at finding an orthogonal matrix \( B \in \mathbb{R}^{J \times J} \) so that it can transform each training frame into a sparse vector. The learning problem is formulated as follows:

\[
\min_{\{e_i\}_{i=1}^N} \sum_{i=1}^N \|B^d m_i^d - e_i^d\|_2^2
\]

subject to \( B^d B^d^T = B^d^T B^d = I \),

\[
\|e_i^d\|_0 \leq P, \quad (6)
\]

where the \( \ell_0 \)-norm \( \|\cdot\|_0 \) counts the number of non-zero entries in the transform coefficient of the \( i \)-th training sample, and \( P \) is the user-specified parameter controlling the sparsity in \( e_i \). \( I \in \mathbb{R}^{J \times J} \) is the identity matrix.

The problem in Equation (6) is non-convex due to the non-convex constraints. We develop an alternating iterative method, which alternatively solves the following two subproblems until convergence:

4.2.1 The Sparse Vector Subproblem

With fixed \( B^d \), let \( g_i^d = B^d m_i^d \). The sparse vector subproblem is equivalent to the summation of multiple independent univariate minimization problems, and the \( i \)-th one is expressed as

\[
\min_{\{e_i^d\}} \|g_i^d - e_i^d\|_2^2 \quad \text{subject to} \quad \|e_i^d\|_0 \leq P. \quad (7)
\]

Obviously, the minimization is achieved only when \( e_i^d \) contains the \( P \) largest entries (in magnitude) of \( g_i^d \), which are at the corresponding locations. Therefore, we can compute \( e_i^d \) by setting the \((J-P)\) smallest (in magnitude) entries of \( B^d m_i^d \) to zero:

\[
e_i^d = T \left( g_i^d, J - P \right), \quad (8)
\]

where \( T \) denotes the truncating operation.

4.2.2 The Orthogonal Matrix Subproblem

Given fixed sparse vectors \( e_i^d \), \( i = 1, \ldots, N \), let us denote \( E^d = [e_1^d, \ldots, e_N^d] \) the matrix representation. The orthogonal matrix subproblem is

\[
\min_{B^d} \|B^d M^d - E^d\|_F^2
\]

subject to \( B^d B^d^T = B^d^T B^d = I \),

\[
(9)
\]

where \( \|\cdot\|_F \) is the Frobenius norm of matrix and \( M^d \) is the matrix representation of all training frames. Observe that

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

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

where \( \text{Tr} \) is the matrix trace.

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

\[
\max_{B^d} \text{Tr} \left( B^d M^d E^d^T \right), \quad \text{subject to} \quad B^d B^d^T = B^d^T B^d = I. \quad (10)
\]

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

Then we can rewrite the objective function as

\[
\text{Tr} \left( \widetilde{B}^d M^d {E^d}^T \right) = \text{Tr} \left( \widetilde{B}^d \tilde{U}^d S^d \tilde{V}^d \right) = \text{Tr} \left( \tilde{B}^d \tilde{U}^d S^d \right),
\]

where \( \tilde{B}^d = \widetilde{B}^d \) is still an orthogonal matrix.

Since \( S^d \) is a diagonal matrix, maximizing (10) is equivalent to maximizing the diagonal entries of \( B^d \tilde{U}^d \).

With Cauchy-Schwartz inequality, the \( i \)-th diagonal entry of \( B^d \tilde{U}^d \) is

\[
\sum_{j=1}^{J} B^d_{ij} \tilde{U}^d_{ji} \leq \sqrt{\sum_{j=1}^{J} B^d_{ij}^2 \sum_{j=1}^{J} \tilde{U}^d_{ji}^2} = 1.
\]

The last equation comes from the fact that both \( B \) and \( \tilde{U} \) are orthogonal matrices.

Therefore, the objective function in (10) is maximized when \( B^d \tilde{U}^d = I \), leading to

\[
B^d \approx \tilde{V}^d \tilde{U}^d. \tag{11}
\]

Algorithm 1 shows the pseudocode of the LOT algorithm. In each iteration, the truncating operation (line 4) and matrix multiplication (lines 6) take \( J \log J \) time and \( 2NJ^2 \), respectively. Singular value decomposition has an \( O(J^3) \) time complexity. Putting it all together, the time complexity of Algorithm 1 is \( O(KNJ^2 + KNJ \log J + KJ^3) \). 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).

**Algorithm 1 Computing LOT Bases for Mocap Data**

**Input:** training samples \( \{m_i\}_{i=1}^N \), the sparsity parameter \( P \) and the maximum number of iterations \( K \)

**Output:** the orthogonal matrix \( B \)

1: initialize \( B \) using an orthogonal matrix (e.g., DCT or DWT bases)
2: for \( \text{iter} \leftarrow 1 : K \) do
3: for \( i \leftarrow 1 : N \) do
4: update \( e_i^d \) using (8)
5: end for
6: factor \( M^d {E^d}^T \) using SVD
7: update \( B \) using (11)
8: end for

5 EXPERIMENTAL RESULTS AND DISCUSSION

We implement the proposed schemes in MATLAB and evaluate them on the CMU Mocap Database\(^2\) in which each frame consists of \( J = 31 \) key points sampled at 120 frames per second (fps). Table 1 describes the training and test motion sequences including various motion characteristics and lengths.

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

\[
D = \frac{1}{JP} \sum_{i=1}^{J} \sum_{j=1}^{F} \|p_{i,j} - \hat{p}_{i,j}\|_2. \tag{12}
\]

![Fig. 2. Visualization of the 1D DCT and LOT 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.](image)

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.

**Table 1** Description of training sequences and test sequences.

| Sequence | Size (kB) | Description |
|----------|-----------|-------------|
| 86_02    | 10,617    | walk, squats, run, stretch, jumps, punches, and drinking |
| 56_04    | 6,767     | lists up, wipe window, grab, walk, throw punches, yaw, stretch, jump |
| 15_05    | 22,948    | wash windows, paint, hand signals, dance, dive, twist, boxing |
| 14_08    | 7,825     | jump up to grab |
| 15_04    | 22,149    | dance, the twist, boxing |
| 17_08    | 6,179     | muscular person’s walk |
| 14_10    | 2,783     | jump, hop on one foot |
| 56_07    | 9,420     | yawn, stretch, walk, run, halt |
| 85_12    | 4,499     | jumps, flips, breakdance |
| 86_05    | 8,340     | walking, jumping, punching |

2. http://mocap.cs.cmu.edu/
5.1 Training the LOT Bases

We take sequences “86_02” “56_04”, and “15_05” as the training set, since they consist of various types of motions. It is worth noting that more training frames can generate better performance, but the computational cost also increases.

The LOT 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 Figure 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. Figure 3 also visualizes the bases of 1D DCT and LOT 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 turns, controls the compression performance. In the training process, we set \( P \) to four different values, i.e., 2, 5, 8, and 11, respectively. Then, the learned orthogonal matrices under different \( P \) are tested in the frame- and clip-based frameworks, respectively. For both schemes, four randomly chosen sequences with various motion characteristics and lengths are compressed, and the results are shown in Figure 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 LOT, 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 Figure 5 shows, given the same number of nonzero transformed coefficients, the distortions produced by LOT are consistently much smaller than those of DCT and DWT, meaning that LOT concentrates energy (or spatially decorrelated mocap data) better than DCT and DWT.
5.3 Compression Performance

Figure 6 shows the CR-distortion (CR-D) curves of the frame-based scheme. As Section 5.2 shows, our data-adapted LOT is more superior than 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 [10] in terms of compression performance. We observe that with a relatively high CR, our frame-based scheme can reduce up to 70% distortion of [10].

Figure 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 joint trajectories in a clip still remain smooth and have small variation (due to the small duration), causing the DCT coefficients to

---

Fig. 6. Comparison of compression performance of frame-based methods.

Fig. 7. Compression performance of our clip-based schemes and the state-of-the-art methods, namely, the PCA-RDO method [7] and the MDTT method [8].
5.4 Comparison

In this subsection, we compare our clip-based scheme with only two works, namely PCA-Rate Distortion Optimization (PCA-RDO method) [7], and the equal segmentation case of Mocap Data Tailored Transform (MDTT) method [8], which represent the state-of-the-art. See [7] and [8] for detailed performance evaluation on earlier works [11], [6], [5], [4], [24].

5.4.1 Comparison with the MDTT Method

Both our clip-based algorithm and the MDTT method [8] apply temporal DCT to each joint trajectory for temporal decorrelation, since each joint trajectory is a signal defined on the 1D chain graph, where the 1D DCT is optimal for decorrelation under Gaussian Markov random field assumption. 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 functions. Within our clip-based framework, the LOT 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 to reduce the dimension of transformed coefficients. In contrast, the LOT makes the transform coefficients sparse by quantization, which is more flexible.

From the CR-D curves in Figure 7, we observe the MDTT [8] 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) are 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 remained 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 applications where large latency is tolerated, whereas our method is desired for time-critical applications (regardless of sequence length) such as interaction.

5.4.2 Comparison with the PCA-RDO Method

The PCA-RDO method [7] 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. Vásá and Brunnett [7] 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. As Figures 7(c)(d)(e)(i) show, our clip-based scheme consistently outperforms the PCA-RDO method [7] in terms of compression performance.
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 framework, the user only needs to specify the clip length $L$, which directly controls the compression performance as well as the latency.

5.5 Discussion

In this paper, the compression distortion by our schemes is measured as the Euclidean distance between the positions of original and decompressed joints. The mocap data is always used for skinning animation of a surface mesh. Therefore, it is highly desired to evaluate the performance of mocap compression schemes by measuring the quality of meshes animated using the decompressed mocap data. Váša and Brunnett recently proposed a perceptual-based distortion metric for mocap data. However, the metric is not validated by subjective experiments. Subjective experiments for mocap data were
performed in [6], [23], but they just showed animations of stick figures instead of animations of meshes to users.

6 Conclusion

We have presented two effective frameworks – one is frame-based and the other is clip-based – for compressing human mocap data with low latency. Since mocap data exhibit some unique spatial characteristics, we propose a very effective transform, namely learned orthogonal transform, for reducing the spatial redundancy. Thanks to its data adapted nature, the proposed transform outperforms the commonly used data-independent transforms, such as discrete cosine transform and discrete wavelet transform, in terms of spatial decorrelation. Experimental results show that the proposed frameworks can produce higher compression performance at a lower computational cost than the state-of-the-art methods.

References

[1] T. Capin, I. Pandžič, N. Magnenat-Thalmann, and D. Thalmann, *Avatars in Networked Virtual Environments*. John Wiley & sons, 1999.
[2] M. Gutierrez, F. Vexo, and D. Thalmann, “Controlling virtual humans using pdas,” in Proceedings of the 9th International Conference on Multi-Media Modeling, 2003, pp. 150–166.
[3] S. Chattopadhyay, S. Bhandarkar, and K. Li, “Human motion capture data compression by model-based indexing: A power aware approach,” IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 1, pp. 5–14, Jan 2007.
[4] Q. Gu, J. Peng, and Z. Deng, “Compression of human motion capture data using motion pattern indexing,” Computer Graphics Forum, vol. 28, no. 1, pp. 1–12, 2009.
[5] M. Tournier, X. Wu, N. Courty, E. Arnaud, and L. Reveret, “Motion compression using principal geodesics analysis,” Computer Graphics Forum, vol. 28, no. 2, pp. 355–364, 2009.
[6] I.-C. Lin, J.-Y. Peng, C.-C. Lin, and M.-H. Tsai, “Adaptive motion data representation with repeated motion analysis,” IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 4, pp. 527–538, April 2011.
[7] L. Váša and G. Brunnett, “Rate-distortion optimized compression of motion capture data,” Computer Graphics Forum, vol. 33, no. 2, pp. 283–292, 2014.
[8] J. Hou, L.-P. Chau, N. Magnenat-Thalmann, and Y. He, “Human motion capture data tailored transform coding,” IEEE Transactions on Visualization and Computer Graphics, vol. 21, no. 7, pp. 848–859, 2015.
[9] ——, “Scalable and compact representation for motion capture data using tensor decomposition,” IEEE Signal Processing Letters, vol. 21, no. 3, pp. 255–259, March 2014.
[10] C.-H. Kwak and I. V. Bajic, “Hybrid low-delay compression of motion capture data,” in Proceedings of IEEE International Conference on Multimedia and Expo (ICME), 2011, pp. 1–6.
[11] O. Arikan, “Compression of motion capture databases,” ACM Transactions on Graphics, vol. 25, no. 3, pp. 890–897, 2006.
[12] G. Liu and L. McMillan, “Segment-based human motion compression,” in Proceedings of the ACM SIGGRAPH/Eurographics SCA, 2006, pp. 127–135.
[13] B.-S. Chew, L.-P. Chau, and K.-H. Yap, “A fuzzy clustering algorithm for virtual character animation representation,” IEEE Transactions on Multimedia, vol. 13, no. 1, pp. 40–49, Feb 2011.
[14] M. Alexa and W. Miller, “Representing animations by principal components,” Computer Graphics Forum, vol. 19, no. 3, pp. 411–419, 2000.
[15] Z. Karni and C. Gotsman, “Compression of soft-body animation sequences,” Computers & Graphics, vol. 28, no. 1, pp. 25–34, 2004.
[16] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the h.264/avc video coding standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560–576, July 2003.
[17] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649–1668, Dec 2012.
[18] X. Gu, S. J. Gortler, and H. Hoppe, “Geometry images,” ACM Transactions on Graphics, vol. 21, no. 3, pp. 355–361, 2002.
[19] J. Hou, L.-P. Chau, M. Zhang, N. Magnenat-Thalmann, and Y. He, “A highly efficient compression framework for time-varying 3-d facial expressions,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, no. 9, pp. 1541–1553, Sept 2014.
[20] J. Hou, L. Chau, N. Magnenat-Thalmann, and Y. He, “Compressing 3-d human motions via keyframe-based geometry videos,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 1, pp. 51–62, Jan 2015.
[21] M. Preda, B. Jovanova, I. Arsov, and F. Prêteux, “Optimized mpeg-4 animation encoder for motion capture data,” in Proceedings of the International Conference on 3D Web Technology, 2007, pp. 181–190.
[22] P. Beaudoin, P. Poulin, and M. van de Panne, “Adapting wavelet compression to human motion capture clips,” in Proceedings of Graphics Interface, 2007, pp. 313–318.
[23] A. Firozumanesh, I. Cheng, and A. Basu, “Perceptually guided fast compression of 3-d motion capture data,” IEEE Transactions on Multimedia, vol. 13, no. 4, pp. 829–834, Aug 2011.
[24] M. Zhu, H. Sun, and Z. Deng, “Quaternion space sparse decom- position for motion compression and retrieval,” in Proceedings of the ACM SIGGRAPH/Eurographics SCA, 2012, pp. 183–192.
[25] Z. Zhang and D. Fiorêncio, “Analyzing the optimality of predictive transform coding using graph-based models,” IEEE Signal Processing Letters, vol. 20, no. 1, pp. 106–109, 2013.
[26] A. Safonova, J. K. Hodgins, and N. S. Pollard, “Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces,” ACM Transactions on Graphics, 3, pp. 514–521, 2004.