

# High-fidelity Compression of Dynamic Meshes with Fine Details using Piece-wise Manifold Harmonic Bases

Chengju Chen

State Key Laboratory of Virtual  
Reality Technology and Systems,  
Beihang University  
Beijing, China  
chengjuchina@gmail.com

Qing Xia

State Key Laboratory of Virtual  
Reality Technology and Systems,  
Beihang University  
Beijing, China  
neijiangxiaqing@gmail.com

Shuai Li\*

State Key Laboratory of Virtual  
Reality Technology and Systems,  
Beihang University  
Beijing, China  
Beihang University Qingdao Research  
Institute, Beihang University  
Qingdao  
lishuaiouc@126.com

Hong Qin

Department of Computer Science,  
Stony Brook University  
New York, USA  
qin@cs.stonybrook.edu

Aimin Hao

State Key Laboratory of Virtual  
Reality Technology and Systems,  
Beihang University  
Beijing, China  
ham\_buaa@163.com

## ABSTRACT

Mesh-based animation, usually represented as dynamic meshes with fixed connectivity, is becoming more and more prevalent in movies, games and other graphics applications nowadays, and there is a growing need to compactly store and rapidly transmit these meshes for practical use, especially for those with high-quality geometric details. In this paper, we explore a novel key-frame based dynamic mesh compression method, wherein we apply pose-similarity with spectral techniques to define piece-wise manifold harmonic bases to reduce spatial-temporal redundancy. We first partition the sequence into several clusters with similar poses, and then decompose the meshes in each cluster into primary poses and geometric details using the manifold harmonic bases derived from the extracted key-frame in that cluster. The primary poses can be characterized as linear combinations of manifold harmonic bases, and the geometric details can be recovered by deformation transfer technique. Thus, we only need a small number of key-frames and a few coefficients for compressing dynamic meshes, which saves a significant amount of storage comparing to traditional methods in which bases are stored explicitly. Furthermore, we apply a second-order linear prediction coding to the harmonic coefficients to further reduce the temporal redundancy. Our extensive experiments and evaluations on various datasets have manifested that our novel method could obtain a high compression ratio while preserving

high-fidelity geometry details and guaranteeing limited human perceived distortion rate simultaneously.

## CCS CONCEPTS

• Computing methodologies → Animation;

## KEYWORDS

Animated Mesh Compression, Manifold Harmonic Basis(MHB), Linear Prediction Coding, Key Frame Extraction

## ACM Reference Format:

Chengju Chen, Qing Xia, Shuai Li, Hong Qin, and Aimin Hao. 2018. High-fidelity Compression of Dynamic Meshes with Fine Details using Piece-wise Manifold Harmonic Bases. In *CGI 2018: Computer Graphics International 2018, June 11–14, 2018, Bintan Island, Indonesia*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3208159.3208163>

## 1 INTRODUCTION

With the rapid technical advancement in movie and game industry, computer animation is becoming prevalently popular. A digital animation is usually generated by driving a character mesh created by artists in certain manner, and stored as a sequence of dynamic meshes with fixed connectivity and time-varying geometry. Such dynamic meshes, especially those with complex geometric details, require large storage space and transmission bandwidth in practical computer graphics applications. Therefore, high-fidelity compression of dynamic meshes with fine details has gained increasing attention during the past decades.

In essence, the goal of dynamic meshes compression is to find a compact representation of the sequence with a controllable distortion, and one intuitive way is to pursue a set of bases that can well characterize geometry of the meshes. Principle Component Analysis (PCA) has been reused frequently in combination with other compression techniques [1, 12, 26], wherein the meshes are projected onto a few principal orthogonal bases and the compression

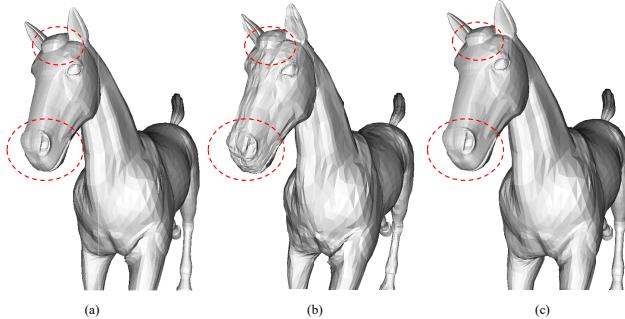
\*Prof. Li is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CGI 2018, June 11–14, 2018, Bintan Island, Indonesia

© 2018 Association for Computing Machinery.  
ACM ISBN 978-1-4503-6401-0/18/06...\$15.00  
<https://doi.org/10.1145/3208159.3208163>

is achieved by removing the bases whose influence are negligible compared with others. Nonetheless, these bases take up to 50% of the encoded data size [27] and inevitably limit the compression ratio of these methods when being applied to meshes with a lot of geometric details or a large number of vertices, so it is necessary to find a compression method which does not need to store bases explicitly; on the other hand, even though these methods achieve high compression rate under traditional limited vertex-based error measurement, such as Karni-Gotsman (KG) error [10], the removal of bases inevitably cause information loss and consequently affect the quality of reconstructed meshes from the perceptual aspect, as shown in Fig. 1. We shall explore a high-fidelity way for dynamic meshes compression, whose reconstructed meshes should be visually plausible which can be evaluated under perceptual metric, such as Spatial-Temporal Edge Difference (STED) [28].

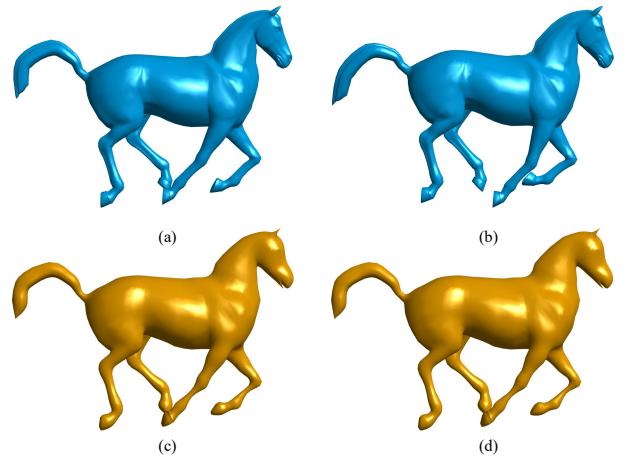


**Figure 1: Comparison with CoDDyaC under the same compression ratio. From left to right: the original frame (a), reconstructed frame using CoDDyaC (b), reconstructed frame using our newly-proposed method (c).**

Manifold harmonic transform [23] projects a 3D mesh onto several Fourier-like function bases derived as eigenfunctions of the Laplace-Beltrami operator, and converts the mesh from geometry domain into frequency domain. Some static mesh compression methods [9, 30] showed that they can reconstruct a static mesh with a small number of low-frequency bases according to the assumption that low-frequency coefficients contribute more to the mesh than the high-frequency ones. In addition, these bases can be obtained by solving an eigenvalue problem of the Laplace-Beltrami operator defined on the mesh itself without the need for explicit storage. However, simply ignoring the high-frequency bases certainly loses geometric details and causes errors similar to those in PCA-based methods. Nonetheless, how to apply the manifold harmonic bases (MHBs) to high-fidelity dynamic meshes compression still remains challenging.

In this paper, we propose a key-frame based framework for dynamic meshes compression with manifold harmonic bases, in which we decompose the sequence into primary poses (low-frequency part) and geometric details (high-frequency part) and compress them separately by exploiting spatial-temporal correlations among data. We first partition the sequence by the notion of *pose similarity* into several small fragments and observe that in each fragment, low-frequency parts of all meshes can be well described by manifold harmonic bases derived from one single mesh within this

fragment. As shown in Fig. 2, we can compress the low-frequency part in this fragment by storing one representative key-frame and a few coefficients instead of many large bases to reduce the spatial redundancy.



**Figure 2: Top row, Horse-gallap sequence’s Frame 28 (a), and Frame 30 (b). Bottom row, reconstructed low frequency part of Frame 28 using 400 MHBs of Frame 28 itself (c), and using 400 MHBs of Frame 30 (d).**

For high-frequency part, we find that geometric details of the meshes within one fragment are almost unchanged, because a mesh sequence is usually created by deforming one static mesh along time axis and geometric details of meshes should be similar to those with similar poses. So we can just store the details of one representative key-frame to remove redundant information and then transfer the details to the reconstructed primary poses in order to reconstruct high quality sequence with a limited distortion. Furthermore, we apply a second-order linear prediction to reduce the temporal redundancy of harmonic coefficients. In particular, the primary contributions of this paper can be summarized as follows:

- We devise a hierarchical framework in which we decompose the sequence into high frequency part and low frequency part and compress them separately to reduce the geometric redundancy in space.
- We apply the notion of pose similarity to extract the key-frames and define the piece-wise manifold harmonic bases to compress the primary poses.
- We apply deformation transfer techniques to preserve the geometry details in order to reconstruct high quality mesh sequence.
- We propose a multi-resolution scheme for complicated mesh sequence which has many details and too many vertices.

## 2 RELATED WORK

All compression schemes aim at exploiting correlations among mesh sequence. In terms of de-correlation techniques, the existing compression methods can be roughly classified into two categories [13]: spatial de-correlation methods and temporal de-correlation

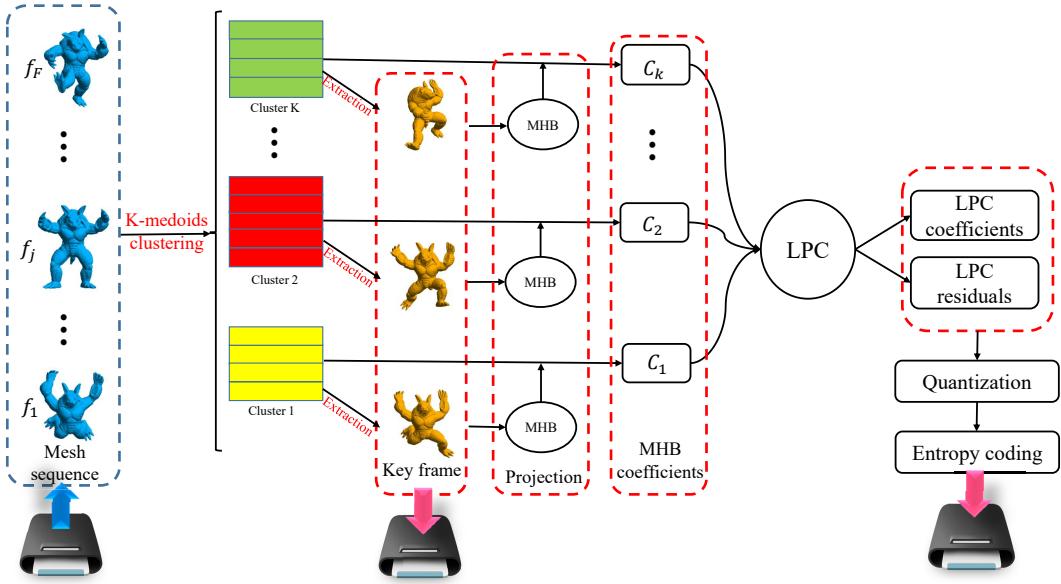


Figure 3: Flowchart of the proposed compression algorithm (encoder).

methods. Actually most methods explore both temporal and spatial redundancy, the two categories are roughly classified according to whether the core idea is benefited more from spatial de-correlation or temporal de-correlation.

**Spatial de-correlation.** The most commonly used method to exploit spatial correlations is *principal component analysis* (PCA) which was firstly introduced to mesh sequence compression by Alexa and Muller [1] in which they projected the sequence data onto a few principal orthogonal bases to represent each frame as a linear combination of these bases. The approach is improved by Lee *et al.* [11] who described how to determine the optimal number of PCA bases. Then Karni *et al.* [10] applied linear prediction coding to further compress PCA coefficients.

Some authors proposed methods that segment frames into meaningful clusters to exploit spatial correlation. Sattler *et al.* [17] first clustered the vertex trajectories and applied PCA to the trajectories of all vertices of a cluster. Mamou *et al.* [14] proposed a skinning approach based on segmentation. The mesh vertices are partitioned into patches whose motion can be accurately described by a 3D affine transform. Then an affine motion model is defined to estimate the frame-wise motion of each patch. Hou *et al.* [6] proposed learned spatial decorrelation transform to transform each frame into a sparse vector to reduce the spatial redundancy and then they [7] used low-rank matrix approximation for data compression.

Some methods encode only a set of key-information and used them to predict the missing information with some spatial predictors. Stefanoski *et al.* [19] proposed a linear predictive compression approach in which patch-based mesh simplification algorithms are applied to derive spatially decomposed layers of each frame so that their method supports spatial scalability. Then scalable predictive coding (SPC) [21] is proposed to support spatial-temporal scalability by decomposing animated meshes in spatial and temporal layers and predicting these layers using the already encoded

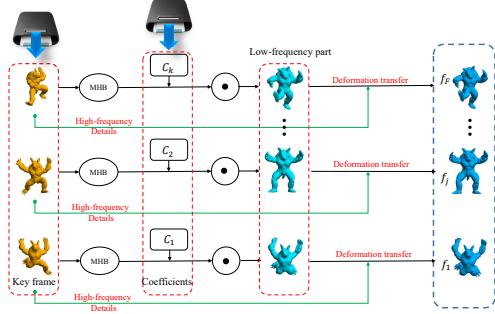
spatio-temporal neighborhood. It is improved by Bici *et al.* [2], who proposed three novel prediction structures based on weighted spatial prediction with its weighted refinement and angular relations of triangles between current and previous frames. Hajizadeh *et al.* [4] proposed a key-frame based technique in which extracted key-frames are then linearly combined using blending weights to predict the vertex locations of the other frames.

**Temporal de-correlation.** PCA can be also used to exploit temporal coherence by applying it to the space of vertex trajectories rather than shapes. Vasa and Skala [26] proposed CoDDyaC algorithm in which they applied PCA to the vertex trajectories to find a minimal number of significant trajectories characterizing the motion of the shape over the sequence. Compared with PCA on shape space, it involves the eigenvalues decomposition of a covariance matrix of  $3F \times 3F$  instead of  $3V \times 3V$ , where  $F, V$  is the number of frames and vertices of the mesh, respectively. Improved results are obtain by predicting PCA bases with an efficient mechanism [27]. Then in [25] they combined CoDDyaC with a novel spatio-temporal predictor and used a discrete geometric Laplacian of average surface to encode the coefficients to achieve a good compression rate. Luo *et al.* [12] applied PCA to temporal clusters based on *pose similarity* to extend the notion of temporal coherence to postural coherence.

Payan and Antonini [16] exploited the temporal coherence by using a temporal wavelet filtering and using a bit allocation process to optimize the resulting wavelet coefficients. A signal-to-noise ratio (SNR) and temporal scalable coding algorithm for 3-D mesh sequences using singular value decomposition (SVD) was proposed by Heu *et al.* [5]. They developed a temporal prediction mode to improve the rate-distortion performance which also supports temporal scalability. In 2003, Ibarria *et al.* [8] introduced two extrapolating space-time predictors: the ELP extension of the Lorenzo predictor and the Replica predictor. With these two predictors they predicted

the position of each vertex  $v$  of frame  $f$  from three of its neighbors in frame  $f$  and from the positions of  $v$  and of these neighbors in the previous frame.

### 3 METHOD OVERVIEW



**Figure 4:** Flowchart of the proposed decompression algorithm (decoder).

Given a sequence of  $F$  triangle meshes  $f_i$ ,  $i = 1, \dots, F$  with fixed connectivity of  $N$  vertices in each frame, a frame  $f_i$  is composed of the vertex coordinates of the mesh which we represent as a column vector:  $f_i = [x_{i,1} \dots x_{i,N} \ y_{i,1} \dots y_{i,N} \ z_{i,1} \dots z_{i,N}]^T$ . The mesh sequence is then represented with a matrix  $A$  whose columns are the frames of the animation,  $A = [f_1 \dots f_F]$ , we assume that the connectivity is encoded once using any state-of-the-art algorithm. The purpose of our method is to compress the time-varying geometry data only. Our method can also be briefly summarized as an encoder shown in Fig. 3 and a decoder shown in Fig. 4. Several main steps involved here are described in details in the following sections.

**Key-frame extraction based on pose-similarity.** As shown in Fig. 3, the first step of the encoder is to group the frames into clusters in which poses are similar to each other and then extract one key-frame in each cluster. Geometry of the frames with similar posture within one cluster lie near in a  $3N$  dimensional space. A Manhattan distance  $\|f_j - f_i\|_1$  is used to measure the pose similarity between two frame vectors  $f_i$  and  $f_j$ . Note that, the frames belonging to the same cluster may not be contiguous.

**Intra-cluster compression with piece-wise MHBs.** The second step of the encoder is to decompose the meshes into high-frequency parts and low-frequency parts. We compress low-frequency part in each cluster by projecting them onto a manifold harmonics basis derived from each key-frame and apply second-order linear predictive coding to encode the MHB coefficients. Note that, the manifold harmonic bases of key-frames are grouped into piece-wise MHBs.

**Decompression with deformation transfer.** In the decoder, we first use the stored key-frames and MHB coefficients to reconstruct the primary poses of the sequence, and then transfer the geometric details to the decompressed primary poses to recovery the high-quality mesh sequence. Instead of storing all high frequency details, we just store the geometry details of key-frames to exploit the spatial correlation.

### 4 KEY-FRAME EXTRACTION BASED ON POSE-SIMILARITY

The first step of key-frame extraction is to divide a long and complicated mesh sequence into some clusters in which geometry of the frames with similar posture lie in a space with much lower dimensionality than the whole sequence. Compressing each cluster separately enables us to reduce the temporal redundancy in order to achieve a higher compression rate and retain a reasonable reconstruction quality at the same time.

Numerous methods have been proposed for pose clustering. In this paper we employ K-medoids clustering to cluster the sequence. Given a mesh sequence  $A$  with dimension  $3N \times F$ , K-medoids clusters the  $F$  frames into  $J$  ( $J < F$ ) clusters ( $S_1, S_2, \dots, S_J$ ):

$$\arg \min_S \sum_{i=1}^J \sum_{f \in S} \|f_j - \mu_i\|_1, \quad (1)$$

where  $\mu_i$  is one frame which has the smallest distance to all other frames in the current cluster. The choice of the number of clusters is an important factor of the compression ratio which will be explained in the next section. Then we select the medoid frame in each cluster as key-frame directly. It is noteworthy that key-frame, which is not only applied to compute the piece-wise MHBs but also applied to recover the high-frequency part of non-key-frames in the decoded process, should be the one that obtains significant posture and representative high-frequency geometry information. The key-frame extraction results for *Walk* are shown in Fig. 5. We store the key-frames using any state-of-the-art static mesh compression methods [18, 24].

Note that we choose K-medoids clustering rather than K-means clustering because in contrast to the latter, K-medoids chooses one of frames as center frame (medoid) rather than the mean of frame vectors in  $S_i$  which maybe produces artifacts, and works with a generalization of the Manhattan Norm to define distance between frames instead of  $L_2$  which is gainful for the key-frame extraction.



**Figure 5:** Extracted key-frames of the *Walk* dataset from [29].

### 5 INTRA-CLUSTER COMPRESSION WITH PIECE-WISE MHBs

Now we have segmented the mesh sequence into clusters such that frames within a cluster share similar postures and extracted

one key-frame for each cluster. We now apply piece-wise manifold harmonic bases to compress the subsequence.

As mentioned above, geometric details of the frames within one cluster are almost unchanged and the geometric primaries lie in a very low dimensional space as the result of pose-similarity clustering. So we devise a hierarchical framework in which we convert the mesh from geometry domain into frequency domain and then the shape space is decomposed into high-frequency and low-frequency domains. We compress these two part separately to reduce spatial redundancy.

In the low-frequency domain, we project the meshes onto a few manifold harmonic bases derived from each key-frame. These bases of the key-frame are derived as eigenvectors of the Laplace-Beltrami operator, which can be computed by solving the following eigenvalue problem [23]:

$$-\mathbf{L}\Phi_k = \lambda\mathbf{M}\Phi_k, \quad (2)$$

where  $\mathbf{M}$  is a  $N \times N$  mass matrix of the key-frame and  $\mathbf{L}$  is a  $N \times N$  matrix which is so-called *cotangent weight*:

$$\mathbf{L}_{ij} = \begin{cases} \cot\alpha_{ij} + \cot\beta_{ij}, & j \in N(i) \\ 0, & j \notin N(i) \\ -\sum_{k \neq i} L_{ik}, & i = j \end{cases}, \quad (3)$$

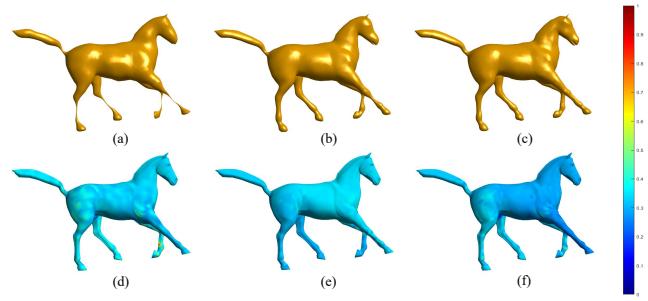
where  $N(i)$  are the vertices adjacent to (neighboring) vertex  $i$ , and  $\alpha_{ij}, \beta_{ij}$  are the angles opposite to edge  $ij$ . We assemble the first  $K$  eigenvectors into a basis matrix  $\mathbf{U} \in R^{N \times K}$ , where each column  $\Phi_k$  is one eigenvector. After performing MHBs for all  $J$  clusters, we get  $J$  new basis matrices  $\{\mathbf{U}_1, \dots, \mathbf{U}_J\}$  which we call piece-wise manifold harmonic bases. Low frequency part of original frame  $\mathbf{f}_j$ , reshaped as  $\mathbf{f}'_j \in R^{N \times 3}$ , from  $\mathbf{S}_i$  is then represented by MHB coefficients:

$$\mathbf{c}_j = \mathbf{U}_i^T * \mathbf{f}'_j \in R^{K \times 3}. \quad (4)$$

Combining the coefficients of frames in the same cluster into  $\mathbf{C}_i$ , we can get coefficient matrices  $\{\mathbf{C}_1, \dots, \mathbf{C}_J\}$  for all clusters. One clear advantage of MHBs is that we can just store or transmit the key-frames and re-compute MHBs in the decompressing process which saves a lot of storage space compared with other PCA-based methods in which they store the PCA bases directly. In addition, manifold harmonic basis captures all the intrinsic properties of the mesh and is invariant to extrinsic shape transformations such as isometric deformations so that the reconstructed meshes have a better shape-fidelity. Note that increasing the number of eigenvectors reduces the distortion, while the total code size might increase due to the additional coefficients as shown in Fig. 6.

In the high-frequency domain, as geometric details of the frames within one cluster are almost unchanged, we can just store the details once to reduce the spatial redundant information. Actually we store full geometry of key-frames which has included the representative geometric details for each cluster in the previous section.

A further compression improvement can be achieved by efficiently encoding the coefficient matrix  $\mathbf{C}$ . In this paper we follow [10] by applying linear predictive coding to compress the MHB coefficient matrix. After LPC, we apply quantization and arithmetic coder to the LPC coefficients and residuals as other compression methods do in order to further reduce the data size.



**Figure 6: Compression with different number of bases.** Top row from left to right: Low-frequency part of Frame 20 of *horsegallap* with 200, 400, 600 bases. Bottom row: Reconstruction errors mapped to blue-red color map with 200, 400, 600 bases.

In summary, the compressed information in our method includes the following two parts:

- the compressed geometry and connectivity of key frames;
- the quantized integers of LPC coefficients and residuals.

## 6 DECOMPRESSION WITH DEFORMATION TRANSFER

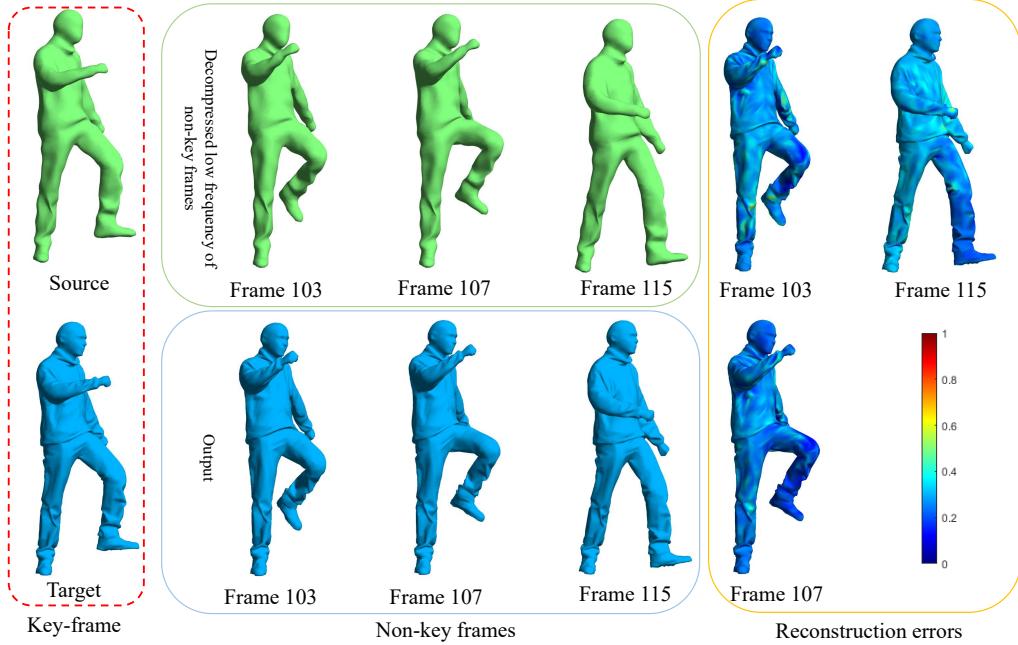
In the decompressing process, we firstly use the LPC coefficients and residuals to retrieve the MHB coefficient matrix  $\mathbf{C}$  (lossless up to quantization). Then with the decompressed key-frames, we re-compute the manifold harmonic bases  $\mathbf{U}_i$  for each cluster. Low-frequency parts of frames in each cluster are reconstructed using corresponding bases and coefficients:

$$\mathbf{f}'_i = \mathbf{U}_i * \mathbf{c}_i. \quad (5)$$

As mentioned above, we only store the geometry of key-frames and low frequency parts of non-key-frames represented by MHB coefficients, so the last step of decompression is to transfer the high-frequency part of key-frames to the low-frequency parts of non-key-frames to recover high-quality mesh sequence via deformation transfer techniques [22]. We first take the low frequency part of key-frame as source mesh and the full key-frame as target mesh. Then the source and target deformations are represented as affine transformations and a correspondence is built here between them. Finally we use this correspondence to map geometry details of key-frames to low frequency parts of non-key frames by solving a constrained optimization as shown in Fig. 7.

Even though we have proposed a key-frame based compression scheme with piece-wise manifold harmonic bases, however computing the spectral bases involves computing the eigenvectors of a  $N \times N$  matrix which will cost much time. Thus, we need a multi-resolution scheme to reduce the space complexities of the sequence to reduce the decompression time.

We firstly use an edge-collapse [3] scheme to coarsen the given mesh sequence with exactly same rule to keep topology consistent across coarse meshes (Ghost) which have around 10000 vertices (actual number depends on mesh complexity), explicitly reducing the information present in the data set. Note that there is a trade-off between space complexities and time, reconstruction error. Then



**Figure 7: Illustration of transferring the high-frequency part of key-frame to the low-frequency parts of non-key frames. The left and middle top row are the decompressed low-frequency parts. The left and middle bottom row are the reconstructed full frames. The right are reconstruction errors mapped to blue-red color map.**

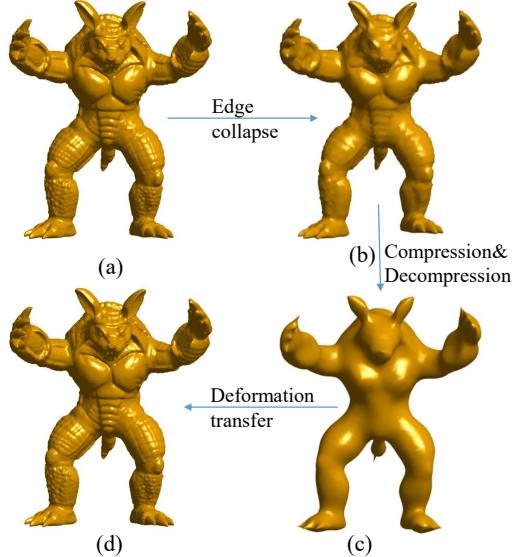
we apply the proposed scheme on the ghost sequence. In the decompressing process, as the topology is modified, we need to build a corresponding relationship between the source and target mesh. Fortunately edge-collapse schemes simplify a mesh by collapsing a few vertices into one. For a vertex on the ghost mesh, we choose the nearest one of these vertices as its corresponding vertex on the original mesh. Then we apply the farthest point sampling method to build sparse correspondences between the target and source meshes and lastly use deformation transfer techniques to recover the high-frequency part. The multi-resolution scheme is shown in Fig. 8.

## 7 EXPERIMENTS AND EVALUATIONS

In this section, we show the experiments of our compression scheme with different mesh sequences which are shown in Table 1. We use bit per vertex per frame (bvpf) to represent the size of data after compression. The overall sequence distortion is measured by the traditional vertex-based error: Karni-Gotsman (KG) error [10] and visual perceptual error: Spatial-Temporal Edge Difference (STED) error [28].

### 7.1 Choice of Compression Parameters

The encoding performance of our scheme is mainly affected by 2 parameters: (1) the number of clusters, (2) the number of manifold harmonic basis. Table 2 shows the compression and distortion results with different numbers of clusters. The optimal number of clusters cannot be defined as a *priori*, but we can roughly estimate a general range according to the pose changing. When the



**Figure 8: Flowchart of the proposed multi-resolution scheme. (a) Original mesh(165954 vertices). (b) Ghost mesh(10002 vertices). (c) Low frequency part of decompressed mesh. (d) Reconstructed mesh.**

similarity of each frame in one cluster is higher, the representation ability of embedded space of MHBs of key-frames is stronger

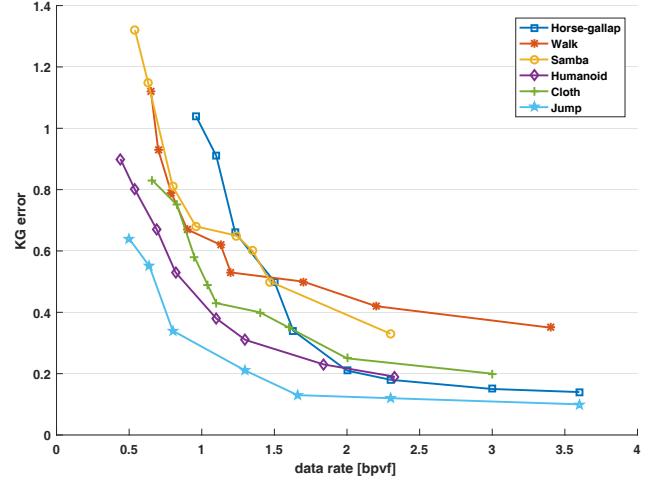
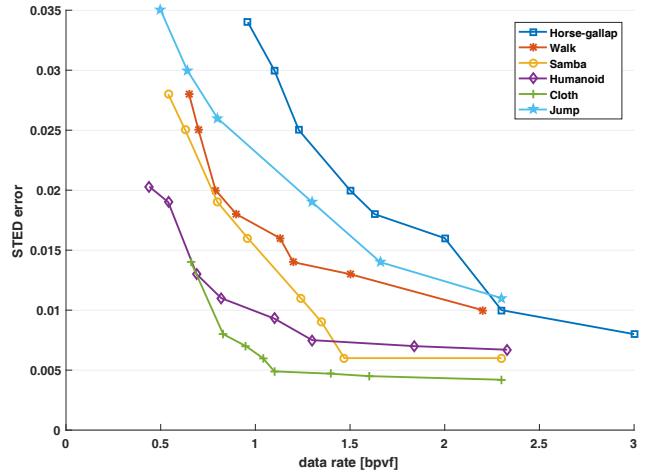
**Table 1: Mesh sequence**

Sequence	Vertices	Triangles	Frames
Cloth	5525	10752	200
Humanoid	7646	15288	154
Horse-gallap	8431	16843	48
Samba	9971	19938	250
Walk	10002	20000	250
Jump	15826	31648	222
Armadillo	165954	331904	81

and the compression performance is better. Taking into the key-postures in the test sequences, experiments show that there is a significant reduction in reconstruction errors with just 2 or 4 clusters for sequences with a few key-postures such as *Horse-gallap* and *Humanoid*, while the complicated sequences such as *Samba* and *Walk* need more clusters. It is obvious that the number of extracted key-frames increases, the KG-error and STED-error reduce while the bvpf increases. Therefore, there is a trade-off between compression ratio and reconstruction error. Table 3 shows the compression and distortion results with different numbers of manifold harmonic bases which determines the threshold of high-frequency domain and low-frequency domain. As the number of MHBs increases, bvpf increases very little, because we just store key-frames and the growing number of MHBs only affects coefficient matrix. It is obvious that the number of bases increases, the KG-error and STED-error reduces which is because more information of low frequency part has transmitted to the decompressing process by embedding the space with more MHBs as shown in Fig. 6. However it needs more time to re-compute MHBs in decompressing process, so there is a trade-off between compression ratio and decompressing time.

**Table 2: Compression performance with different number of clusters**

Sequence	number of clusters	number of bases	bvpf	KG error(%)	STED error
Horse gallap	1	800	0.96	0.84	0.034
	2	800	1.32	0.68	0.024
	4	800	2.20	0.46	0.016
Samba	4	600	0.90	0.61	0.019
	6	600	1.00	0.68	0.016
	10	600	1.35	0.65	0.013
Humanoid	4	600	0.91	0.65	0.014
	6	600	1.11	0.58	0.009
	8	600	1.29	0.57	0.008
Walk	4	600	0.70	0.93	0.025
	8	600	1.13	0.70	0.020
	10	600	1.20	0.63	0.016

**Figure 9: R-D curves for all test sequences measured in KG-error.****Figure 10: R-D curves for all test sequences measured in STED-error.**

## 7.2 Performance with Different Datasets

Fig. 9 and Fig. 10 show typical rate-distortion (R-D) curves under the KG-error and STED-error on all test sequence. It is obvious that the reconstructed error decreases as the data rate increases. As we just store key-frames, our method reduces the required data rate to 0.5 bpvf at 1% distortion rate compared to other traditional PCA-based methods and predictive methods. Our results are seen to be excellent for relatively smooth sequence like *Cloth* and *Jump*. This is due to the low frequencies present the geometry primary more than the sharp parts like fingers or horseshoe. Note that combination of different numbers of bases and clusters may result in different reconstruction errors with the same bvpf. Fig. 11 shows the results of multi-resolution scheme. With the edge collapse scheme the reconstruction error increases while the encoding time decreases

**Table 3: Compression performance with different number of MHB**

Sequence	number of MHBs	number of clusters	bvpf	KG error (%)	STED error	Decompressing time (per frame in seconds)
Horse gallap	600	2	1.25	1.04	0.028	1.04
	800	2	1.32	0.68	0.024	1.72
	1000	2	1.54	0.63	0.023	2.95
Samba	400	6	0.90	0.97	0.019	1.37
	600	6	1.00	0.68	0.016	2.65
	800	6	1.14	0.70	0.015	5.10
Humanoid	300	4	0.90	0.85	0.017	0.66
	600	4	0.91	0.65	0.014	1.51
	800	4	0.95	0.58	0.013	3.41
Walk	400	6	0.93	1.12	0.024	0.88
	600	6	0.98	0.77	0.020	1.97
	800	6	1.08	0.73	0.018	3.27

**Table 4: Encoding computation times (in seconds) of the proposed method for different test sequences**

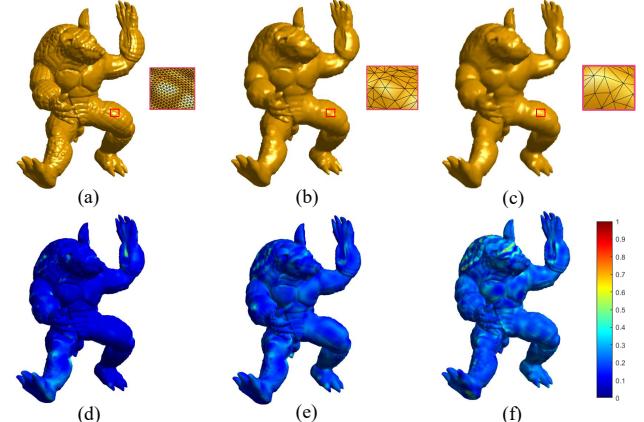
Sequence	Clustering	MHBs	LPC	Total
Cloth	0.349	20.36	49.90	70.60
Horse-gallap	0.438	33.78	21.87	56.09
Samba	0.398	37.83	95.60	133.83
Humanoid	0.366	57.02	61.37	119.2
Walk	0.720	36.86	135.38	172.96
Jump	0.931	58.34	197.90	257.18
Armadillo (10002)	0.228	68.26	43.66	112.48
Armadillo (165954)	3.056	645.35	846.88	1496.28

All measured using 400 bases.

obviously shown in Table 4, so there is a trade-off between distortion rate and decompression time for complex sequence. Table 4 shows the timing cost for the encoding process implemented with Matlab on a PC with Intel Core i7-3770 CPU @ 2.40 GHz. Note that the most time-consuming steps are computing MHBs of key-frames and encoding coefficients which is dependent on the number of vertices.

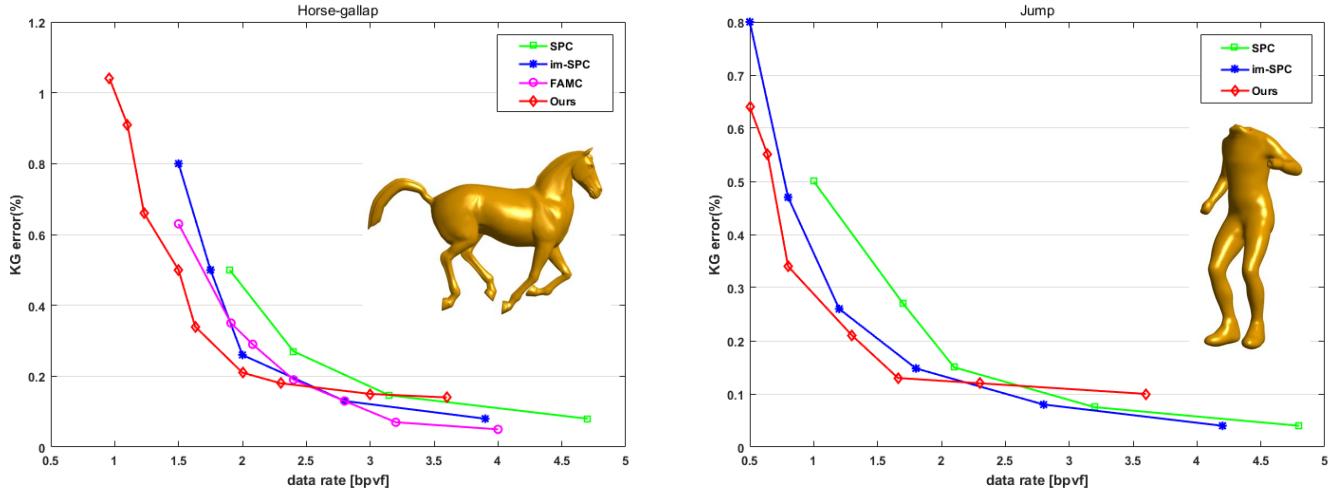
### 7.3 Comparison with Other Methods

We compare with other advanced methods in two aspects, (1) using KG-error to evaluate the accuracy of geometry information; (2) using STED-error to evaluate the perceived distortion which is known as visual loss. Note that we believe that shape fidelity is more important than the accuracy of geometric coordinates. Traditional metrics tend to behave erratically when evaluating the results of Laplacian-based encoding. Instead we use the STED error, since it has been shown to correlate well with perceived distortion.

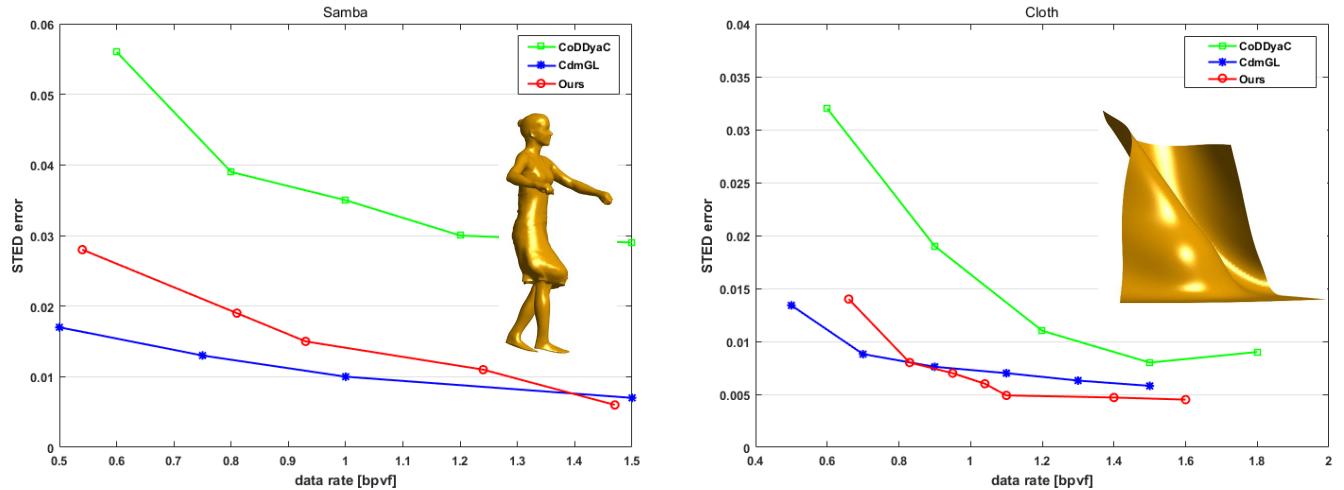


**Figure 11: Multi-resolution scheme for Armadillo.** (a) Original frame with 165954 vertices. (b) Edge collapse to 10002 vertices. (c) Edge collapse to 5002 vertices. (d-f) are the corresponding reconstruction errors.

We first compare the reconstruction accuracy measured by KG-error with previous techniques, namely SPC [19], improved SPC (im-SPC) [2] and MPEG-4 FAMC [20]. Fig. 12 shows the comparison results of *Horse-gallap* and *Jump*. As observed from this figure our algorithm performs better than others in low data rate. That means that inter-frame redundancy can be captured with fewer cluster centers and manifold harmonic bases. With the data rate increases, the KG-error decreases more slowly and becomes almost constant, since we recover the geometry details of the non-key frames via deformation transfer which inevitably lose certain information during transfer process. However we concentrate on obtaining a high compression rate with a limited distortion rate for complicated dynamic mesh sequences, and the results seems satisfactory for this.



**Figure 12:** R-D curves comparison of the proposed method with previous techniques measured by KG-error. Left: *Horse-gallap* sequence. Right: *Jump* sequence.



**Figure 13:** R-D curves comparison of the proposed method with previous techniques measured by STED-error. Left: *Samba* sequence. Right: *Cloth* sequence.

Then we compare the perceived distortion measured by STED-error with CoDDyA C [26] and CdmGL [25] in Fig. 13 with sequence *Cloth* and *Samba*. It shows that our method performs better than CoDDyA C which is a traditional PCA-based method which is also shown in Fig. 1. That is because we can reconstruct the perfect geometry primary with MHBs which occupy an important component in terms of human vision. CdmGL which uses laplacian weights of average shape to encode the delta trajectories, is better than ours in the low data rate. However our scheme is better than CdmGL in the high data rate. That is because we use more Laplacian matrix of key-frames to capture the intrinsic properties of sequence while CdmGL just store one average shape.

## 8 CONCLUSION AND DISCUSSION

In this paper, we have presented a key-frame based method with piece-wise manifold harmonic bases to compress complex mesh sequence. We explored intrinsic geometry properties of sequence with MHBs to reduce spatial redundancy by decomposing the shape space into high frequency part and low frequency part and compressing them separately. Then we applied the notion of pose-similarity to reduce the geometric redundancy along the time axis. We extended the traditional spectral methods to piece-wise manifold harmonic bases to compress the low frequency part of non-key-frames and applied the deformation transfer techniques to recover geometry details of non-key frames. Compared with PCA-based

methods and other predictive methods, we not only eliminated the need for an explicit storage bases to improve the compression ratio, but also achieved better shape fidelity which we strongly believe is more important than the accuracy of geometric coordinates by way of a piece-wise MHBs.

**Limitations and future work.** A possible limitation arises when dealing with frames which have many sharp protrusions, which means we need more bases to reconstruct even the low frequency part subject to a limited reconstruction error and of course the computation cost and compression rate will increase in such cases. One possible solution is to partition the meshes into a few patches with simpler structure and compress these patches separately, or replace the globally defined manifold harmonic bases with the compressed manifold modes (CMM) [15] which has local support in contrast. However, the feasibility and the broader impacts of these solutions will need further investigation in the future. In addition, the parameters in our framework, such as the number of clusters and the number of harmonic bases, are all empirically set according to adequate experiments. Adaptively determining these parameters based on the sequence itself will make our method more practical and adaptable in a wider range of graphics applications.

## ACKNOWLEDGMENT

This research is supported in part by National Natural Science Foundation of China (No. 61672077, 61672149 and 61532002), Applied Basic Research Program of Qingdao (No. 161013xx), National Science Foundation of USA (No. IIS-0949467, IIS-1047715, IIS-1715985, and IIS-1049448), the capital health research and development of special 2016-1-4011, and the Excellence Foundation of BUAA for PhD Students (No. 2017043).

## REFERENCES

- [1] Marc Alexa and Wolfgang Müller. 2000. Representing animations by principal components. In *Computer Graphics Forum*, Vol. 19. Wiley Online Library, 411–418.
- [2] M Oguz Bici and Gozde B Akar. 2011. Improved prediction methods for scalable predictive animated mesh compression. *Journal of Visual Communication and Image Representation* 22, 7 (2011), 577–589.
- [3] Michael Garland and Paul S Heckbert. 1997. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 209–216.
- [4] Mohammadali Hajizadeh and Hossein Ebrahimpour. 2016. Predictive compression of animated 3D models by optimized weighted blending of key-frames. *Computer Animation and Virtual Worlds* 27, 6 (2016), 556–576.
- [5] Jun-Hee Heu, Chang-Su Kim, and Sang-Uk Lee. 2009. SNR and temporal scalable coding of 3-D mesh sequences using singular value decomposition. *Journal of Visual Communication and Image Representation* 20, 7 (2009), 439–449.
- [6] Junhui Hou, Lap-Pui Chau, Nadia Magnenat-Thalmann, and Ying He. 2016. Low-latency compression of mocap data using learned spatial decorrelation transform. *Computer Aided Geometric Design* 43 (2016), 211–225.
- [7] Junhui Hou, Lap-Pui Chau, Nadia Magnenat-Thalmann, and Ying He. 2017. Sparse low-rank matrix approximation for data compression. *IEEE Transactions on Circuits and Systems for Video Technology* 27, 5 (2017), 1043–1054.
- [8] Lawrence Ibarria and Jarek Rossignac. 2003. Dynapack: space-time compression of the 3D animations of triangle meshes with fixed connectivity. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 126–135.
- [9] Zachi Karni and Craig Gotsman. 2000. Spectral compression of mesh geometry. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 279–286.
- [10] Zachi Karni and Craig Gotsman. 2004. Compression of soft-body animation sequences. *Computers & Graphics* 28, 1 (2004), 25–34.
- [11] Pai-Feng Lee, Chi-Kang Kao, Juin-Ling Tseng, Bin-Shyan Jong, and Tsong-Wuu Lin. 2007. 3D animation compression using affine transformation matrix and principal component analysis. *IEICE TRANSACTIONS on Information and Systems* 90, 7 (2007), 1073–1084.
- [12] Guoliang Luo, Frederic Cordier, and Hyewon Seo. 2013. Compression of 3D mesh sequences by temporal segmentation. *Computer Animation and Virtual Worlds* 24, 3-4 (2013), 365–375.
- [13] Adrien Maglo, Guillaume Lavoué, Florent Dupont, and Céline Hudelot. 2015. 3d mesh compression: Survey, comparisons, and emerging trends. *ACM Computing Surveys (CSUR)* 47, 3 (2015), 44.
- [14] Khaled Mamou, Titus Zaharia, and Françoise Prêteux. 2006. A skinning approach for dynamic 3D mesh compression. *Computer Animation and Virtual Worlds* 17, 3-4 (2006), 337–346.
- [15] Thomas Neumann, Kiran Varanasi, Christian Theobalt, Marcus Magnor, and Markus Wacker. 2014. Compressed manifold modes for mesh processing. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 35–44.
- [16] Frédéric Payan and Marc Antonini. 2007. Temporal wavelet-based compression for 3D animated models. *Computers & Graphics* 31, 1 (2007), 77–88.
- [17] Mirko Sattler, Ralf Sarlette, and Reinhard Klein. 2005. Simple and efficient compression of animation sequences. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 209–217.
- [18] Olga Sorkine, Daniel Cohen-Or, and Sivan Toledo. 2003. High-Pass Quantization for Mesh Encoding. In *Symposium on Geometry Processing*, Vol. 42.
- [19] Nikolce Stefanoski, Xiaoliang Liu, Patrick Klie, and Jörn Ostermann. 2007. Scalable linear predictive coding of time-consistent 3d mesh sequences. In *3DTV Conference, 2007*. IEEE, 1–4.
- [20] Nikolce Stefanoski and Jörn Ostermann. 2008. Spatially and temporally scalable compression of animated 3D meshes with MPEG-4/FAMC. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2696–2699.
- [21] Nikolce Stefanoski and Jörn Ostermann. 2010. SPC: fast and efficient scalable predictive coding of animated meshes. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 101–116.
- [22] Robert W Sumner and Jovan Popović. 2004. Deformation transfer for triangle meshes. In *ACM Transactions on Graphics (TOG)*, Vol. 23. ACM, 399–405.
- [23] Bruno Vallet and Bruno Lévy. 2008. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 251–260.
- [24] Libor Váša and Guido Brunnett. 2013. Exploiting connectivity to improve the tangential part of geometry prediction. *IEEE transactions on visualization and computer graphics* 19, 9 (2013), 1467–1475.
- [25] Libor Váša, Stefano Marras, Kai Hormann, and Guido Brunnett. 2014. Compressing dynamic meshes with geometric laplacians. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 145–154.
- [26] Libor Váša and Václav Skala. 2007. Cuddyac: Connectivity driven dynamic mesh compression. In *3DTV Conference, 2007*. IEEE, 1–4.
- [27] Libor Váša and Václav Skala. 2009. COBRA: Compression of the basis for PCA represented animations. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 1529–1540.
- [28] Libor Váša and Václav Skala. 2011. A perception correlated comparison method for dynamic meshes. *IEEE transactions on visualization and computer graphics* 17, 2 (2011), 220–230.
- [29] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. 2008. Articulated mesh animation from multi-view silhouettes. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 97.
- [30] Ming Zhong and Hong Qin. 2014. Sparse approximation of 3D shapes via spectral graph wavelets. *The Visual Computer* 30, 6–8 (2014), 751–761.