

Special Section on SMI 2019

## Denoising of dynamic 3D meshes via low-rank spectral analysis

Gerasimos Arvanitis<sup>a,\*</sup>, Aris S. Lalos<sup>b</sup>, Konstantinos Moustakas<sup>a</sup>

<sup>a</sup>Electrical and Computer Engineering, University of Patras, Greece

<sup>b</sup>Industrial Systems Institute, "ATHENA" Research Center, Patras, Greece



### ARTICLE INFO

#### Article history:

Received 12 March 2019

Revised 22 May 2019

Accepted 22 May 2019

Available online 28 May 2019

#### Keywords:

Spectral denoising via RPCA

Dynamic noisy 3D meshes

Laplacian matrix decomposition

Denoising of graph fourier transform

### ABSTRACT

Recently, the new generation of different 3D scanner devices (e.g., conoscopic holography, structured light, photometric systems, etc.) has attracted a lot of attention due to their ability to provide more reliable results. The easiness of capturing real 3D objects has created revolutionary trends in many areas (e.g., gaming, prominence of heritage, military, medicine, etc.) and has significantly increased the interest for static and dynamic 3D models. However, despite the technological evolution of the 3D acquisition devices, there are still limitations, deteriorating the quality of the generated results (e.g., noise, outliers, and other abnormalities). These issues need to be addressed before the 3D models are used by other applications (such as segmentation, object recognition, tracking, etc.). In this paper, we introduce a novel method which exploits similarities at the spectral frequencies of individual meshes in soft or rigid body 3D animations. The noise is mainly distributed over high frequencies, while the spectrum of the graph Fourier transform of sequential meshes in a 3D animation, exhibits a low-rank which can be effectively exploited using robust principal component analysis (RPCA). Extensive evaluation studies, carried out using a variety of different arbitrarily complex 3D animations and noise patterns, verify that the proposed technique achieves plausible denoising results despite the constraints posed by arbitrarily motion scenarios.

© 2019 Elsevier Ltd. All rights reserved.

### 1. Introduction

Despite the rapid advancements in applications of 3D mesh and point cloud processing, little attention has been given in the area of dynamic 3D mesh denoising. The main reason is that each frame of a dynamic sequence can be considered as an individual mesh which can be handled separately. Although such an approach would result in an efficient exploitation of the spatial properties [1], it completely neglects temporal coherences, ignoring a crucial factor to achieve higher reconstruction quality.

Motivated by this trend, we developed a simple and robust method for denoising noisy 3D mesh sequences. More specifically, the proposed approach is capable of exploiting both small and large scale geometric features at significantly lower computational complexity as compared to denoising approaches that are applied to each frame independently. The main contributions of this work are:

- The fast execution time, achieved also when processing in dense meshes. This remarkable low-computational complexity is attributed to the use of fast ortho-normalization approaches for tracking the graph Fourier subspaces.

- The preservation of geometric features, by exploiting the low-rank spectral properties of the graph Fourier coefficients.
- All the used parameters are pre-defined, based on our experimental analysis, so the users do not need to search for ideal values of parameters per model.
- Despite the theoretical simplicity of the proposed approach, extensive simulation studies show that it can accurately denoise dynamic 3D meshes which share the same connectivity.

Possible applications of the proposed approach include: (i) denoising to improve compression efficiency of state-of-the-art dynamic mesh compression approaches [2], (ii) denoising after the shape completion of motion captured animation [3–5] and (iii) removal of skinning artifacts on 3D frames captured using joint skeleton tracking [6].

The rest of this paper is organized as follows: Section 2 presents related work and prior art in detail. Section 3 presents an overview of our approach, describing in details the workflow of the proposed method. Section 4 presents the experimental results showing the effectiveness of our methods in comparison with other state-of-the-art methods and finally, Section 5 draws the conclusions and we discuss limitations and future directions.

\* Corresponding author.

E-mail addresses: [arvanitis@ece.upatras.gr](mailto:arvanitis@ece.upatras.gr) (G. Arvanitis), [lalos@isi.gr](mailto:lalos@isi.gr) (A.S. Lalos), [moustakas@ece.upatras.gr](mailto:moustakas@ece.upatras.gr) (K. Moustakas).

## 2. Previous works

3D mesh denoising (static or dynamic) is a vital pre-processing step which usually takes place before other more complicated processes (e.g., transmission, segmentation, deforming, compression, etc.) occur. These processes require fully denoised 3D models in order to provide accurate and high-quality reconstruction results. Without a doubt, a lot of works have been presented in the area of 3D mesh denoising. However, despite the significant good results that some of them provide (Zhang et al. [7], Sun et al. [8], Lu et al. [9], Arvanitis et al. [10]) many artifacts are still perceivable, indicating the need for using more sophisticated approaches.

A strict constraint of many works that are not always valid, is the assumption that the noise, affecting the surface of the 3D object, has a Gaussian distribution. This assumption in many cases is not satisfied since in several real-life applications the type and the form of noise have different characteristics (e.g., staircase effect, outliers, devices noise, etc.). Another severe limitation is the fact that many state-of-the-art methods utilize different parameters for each model (He and Schaefer [11], Yadav et al. [12], Wei et al. [13]) and only some parameter-free approaches Wang et al. [14] have been proposed, which also have limitations mainly because they rely on a large dataset for the training process. At the following, we summarize the most important categories that can be found in the denoising literature:

*Bilateral filtering and geometrical features-preserving approaches.* The majority of the state-of-the-art approaches are trying to denoise a mesh taking advantage of spatial similarities of the noisy object's surface. The most popular technique is the bilateral filtering (Fleishman et al. [15], Zheng et al. [16], Jones et al. [17]). Wang et al. [18] proposed an approach for mesh denoising by using a combination of bilateral filtering, feature recognition, anisotropic neighborhood searching, surface fitting and projection techniques. Despite the good results that this method achieves, it also requires the execution of many processes, making the method complex. Lu et al. [9] presented an approach for robust feature-preserving mesh denoising. The method first estimates an initial mesh, then performs feature detection, identification and grouping, and finally, iteratively updates vertex positions based on the constructed feature edges. Centin and Signoroni [19] introduced a feature-preserving denoising technique. They proposed a two stage filtering where the normal filtering is guided by a scale-invariant saliency measure while the vertex filtering is subject to metric accuracy bounds related to the original surface. The main common limitation of the aforementioned approaches is the utilization of different parameters for different models.

*Sparse optimization approaches.* Another important and very popular category for mesh denoising is the sparse optimization methods. Zeng et al. [20] proposed a graph Laplacian regularization based 3D point cloud denoising algorithm. To utilize the self-similarity among surface patches, they adopted the low dimensional manifold prior, and collaboratively denoise the patches by minimizing the manifold dimension. Dinesh et al. [21] proposed to apply graph total variation to the surface normals of neighboring 3D points as regularization. This leads naturally to a  $l_2 - l_1$ -norm objective function, which can be optimized elegantly using ADMM and nested gradient descent. Despite their accurate reconstruction, their increase computational complexity renders them improper for real-time applications.

*Tensor voting approaches.* Wei et al. [22] proposed an approach which exploits the synergy when facet normals and quadric surfaces are integrated to recover a piecewise smooth surface, while the existing mesh denoising techniques focus only on either the first-order features or high-order differential properties. However, they adopt a cascaded operation, which is time-consuming for large models.

*Spectral processing approaches.* Signal processing approaches have played an important role in many applications related to 3D surfaces as demonstrated by Taubin [23], Rustamov and Raif [24] and Zhang et al. [25]. Regarding denoising, Mattei and Castrodad [26] proposed a technique for the restoration of noisy point clouds using a Moving RPCA technique that is based on low-rank and sparse modeling tools. However, this method is time-consuming and the estimated normals are not oriented consistently. Pauly and Gross [27] presented a spectral processing pipeline that extends standard Fourier techniques to general point-sampled geometry. This approach operates directly on points and normals, requiring no vertex connectivity information. Rosman et al. [28] proposed an approach for patch-collaborative spectral denoising of surfaces combining similar patches from the denoised surface. Beltrami operator is used to selectively smooth the surface while preserving sharp surface features. Schall et al. [29] introduced a similarity-based neighborhood filtering technique for static and dynamic data, introducing a non-local similarity measure which determines the resemblance of two points on a surface.

*Data-driven approaches.* Most of the aforementioned categories assume that the noise, affecting the quality of the 3D object, has a Gaussian distribution. This assumption is far from real-life applications in which the type and the form of noise are much more different (staircase effect, outliers, etc.) of this simplified assumption. Only a few parameter-free approaches have appeared providing good results (data-driven methods Wang et al. [14], Remil et al. [30]), but not without limitations mainly because of the large dataset for the training process that they require making them very time-consuming.

In this work, we present a novel approach which investigates the problem of 3D animated sequence denoising from a different view (i.e., spectral denoising) making it distinctive from all of the aforementioned categories. To the best of our knowledge, this is the first time that a method performs denoising on a 3D animated sequence not directly using the geometric information (i.e., points, normals, etc.) of a 3D object but processing the corresponding graph Fourier coefficients.

## 3. Denoising of dynamic 3D meshes

In this section, we present our assumptions and the mathematical background which is necessary for the rest part of the article. We also discuss in detail any step of the process. In Fig. 1, the framework of the proposed approach is briefly presented.

### 3.1. Preliminaries of static and dynamic 3D meshes

Let us assume the existence of a sequence of  $n$  static meshes  $\mathcal{M}_i \in \mathbb{R}^{k \times 3}$  so that  $\mathbf{A} = [\mathcal{M}_1; \mathcal{M}_2; \dots; \mathcal{M}_n]$  representing a dynamic 3D mesh  $\mathbf{A}$ . Each static mesh consists of  $k$  vertices represented as a matrix of vertices  $\mathbf{V} = [\mathbf{v}_1^T; \mathbf{v}_2^T; \dots; \mathbf{v}_k^T] \in \mathbb{R}^{k \times 3}$  in a 3D coordinate space, where  $\mathbf{v} = [v_x \ v_y \ v_z] \in \mathbb{R}^{1 \times 3}$  indicates a vertex. Each  $j$  face  $f_j = \{\mathbf{v}_{j1} \ \mathbf{v}_{j2} \ \mathbf{v}_{j3}\}$ ,  $\forall j = 1, \dots, k_f$  constitutes a triangle (i.e., the basic surface) and  $\mathbf{v}_{j1}$ ,  $\mathbf{v}_{j2}$  and  $\mathbf{v}_{j3}$  define the indices of the corresponding vertices. The vertices of a noisy mesh  $\tilde{\mathcal{M}}_i$  satisfy the following identity:

$$\tilde{\mathbf{v}}_i = \mathbf{v}_i + \tilde{\mathbf{z}}_i, \quad \forall i = 1, \dots, k \quad (1)$$

where  $\mathbf{v}_i$  are the noise free vertices and  $\tilde{\mathbf{z}}_i$  represents a  $1 \times 3$  noise vector (e.g., with distribution  $\mathcal{N}(0, \sigma)$  in the case of Gaussian noise).

### 3.2. Spectrum of a graph

The spectrum of a graph is defined in terms of the eigenvalues and eigenvectors of the Laplacian matrix  $\mathbf{L}$ . The Laplacian matrix

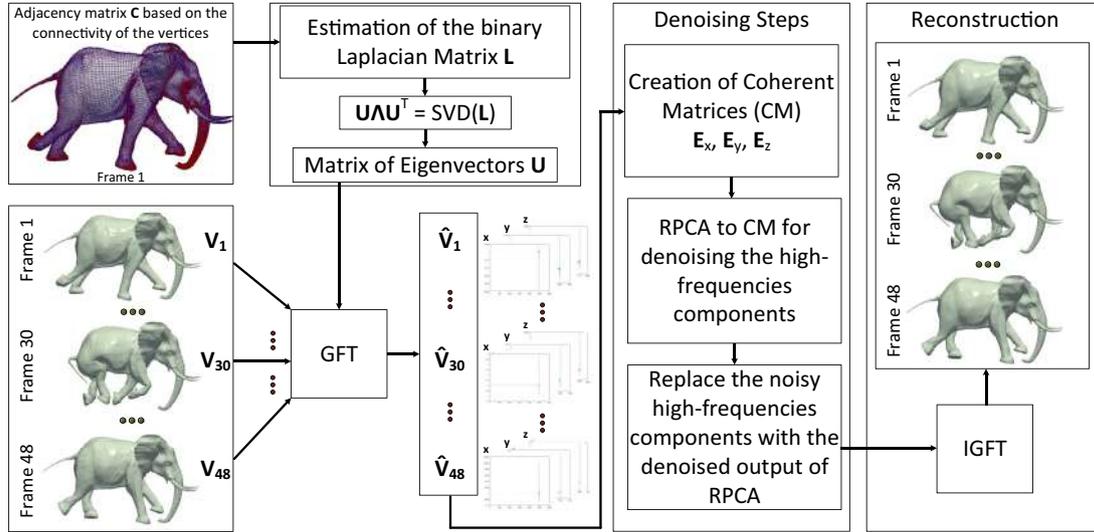


Fig. 1. Steps of the proposed framework for denoising of dynamic 3D meshes via spectral low-rank matrix analysis.

of a graph  $\mathcal{G} = (\mathcal{E}, \mathcal{V}, \mathcal{F})$ , assuming that a set of edges  $\mathcal{E}$  can be directly derived from a set of  $\mathcal{V}$  and the indexed faces  $\mathcal{F}$  of the mesh, can be defined as:

$$\mathbf{L} = \mathbf{D} - \mathbf{C} \quad (2)$$

where  $\mathbf{C} \in \mathbb{R}^{k \times k}$  is the binary connectivity (or adjacency) matrix of the mesh with elements:

$$\mathbf{C}_{ij} = \begin{cases} w_{ij} = 1 & \text{if } i, j \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

and  $\mathbf{D} = \text{diag}\{d_1, d_2, \dots, d_k\}$  is a diagonal matrix with  $d_i = \sum_{j=1}^k \mathbf{C}_{ij}$ . The decomposition of the Laplacian matrix  $\mathbf{L}$  is estimated according to:

$$\mathbf{L} = \mathbf{U}\mathbf{A}\mathbf{U}^T \quad (4)$$

where  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k]$  is an orthonormal matrix with the eigenvectors and  $\mathbf{A} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_k\}$  is a diagonal matrix with the corresponding eigenvalues. The eigenvectors and eigenvalues of the Laplacian matrix  $\mathbf{L}$  provide a spectral interpretation of the graph signals. The GFT of the  $i$ th mesh/frame  $\mathcal{M}_i$ , represented by the matrix of vertices  $\mathbf{V}_i$ , is defined as its projection onto the eigenvalues of the graph, according to:

$$\hat{\mathbf{V}}_i = \mathcal{T}(\mathbf{V}_i) = \mathbf{U}^T \mathbf{V}_i, \quad \forall i = 1, \dots, n \quad (5)$$

where  $\hat{\mathbf{V}} \in \mathbb{R}^{k \times 3}$  is a matrix representing the GFT of the matrix of vertices  $\mathbf{V}$  and  $\mathcal{T}(\cdot)$  represents the GFT function. We can easily observe that the number of components of a GFT matrix is equal to the number of vertices of the mesh. Correspondingly, the inverse GFT (IGFT) of each  $i$  frame is given by:

$$\mathbf{V}_i = \mathcal{T}^{-1}(\hat{\mathbf{V}}_i) = \mathbf{U}\hat{\mathbf{V}}_i, \quad \forall i = 1, \dots, n \quad (6)$$

where  $\mathcal{T}^{-1}(\cdot)$  represents the IGFT function. At any noisy mesh  $\tilde{\mathcal{M}}_i$ , the information corresponds to the shape of the mesh lies in a low dimensional subspace of size  $m$ , while noise usually has a flat spectrum that is easily identifiable at the  $k - m$  higher frequencies in which the small-scale features are also apparent. Previous works of Arvanitis et al. [10], Vallet and Lévy [33] and Lalos et al. [34] suggest performing denoising or smoothing of noisy 3D meshes by removing the high-frequency components in which the component of noise lie. Despite the fact that these works effectively remove the noise of a GFT representation, they also inevitable remove high-frequency components representing small-scale features. One of the contributions of this approach is that these components are not removed but properly processed in order to preserve the small-scale features.

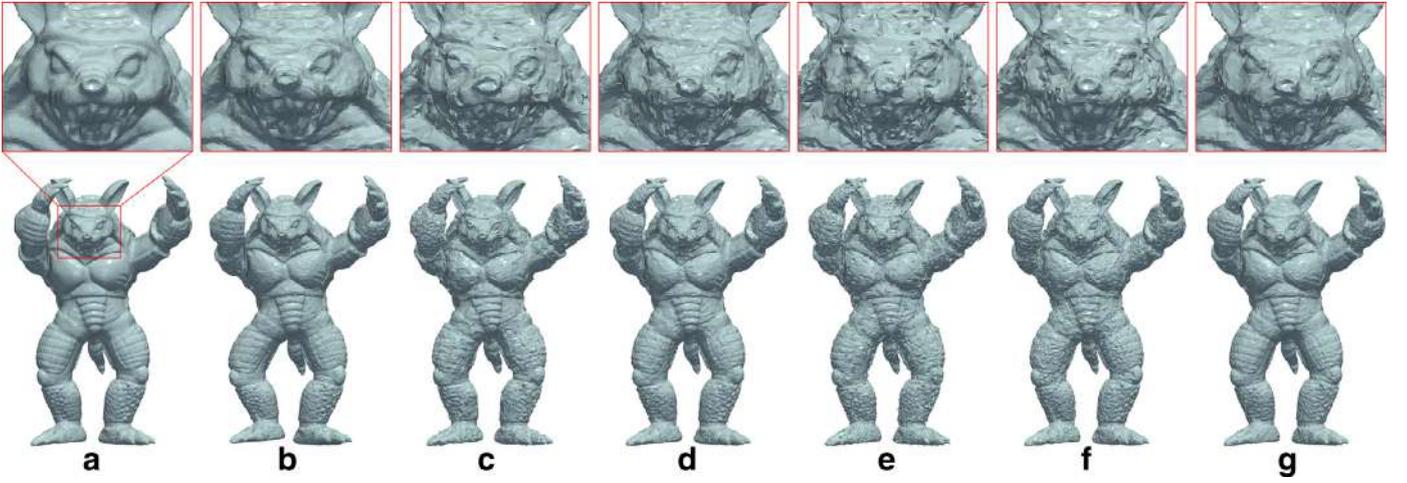
### 3.3. Noisy meshes

The most common types of artifacts introduced in 3D meshes due to scanning operations are: (i) non-uniform sampling, (ii) Gaussian and/or (iii) impulsive noise. Surfaces with holes that can be simulated using (iv) temporal and/or (v) spatial masks. Another type of processing artifact is the noise caused by (vi) lossy compression approaches (e.g., codyac by Vasa and Skala [31], FAMC-DCT by Mamou et al. [32]) or (vii) communication failures (network error). In Fig. 2 we present an example of a 3D mesh (Armadillo) affected by different types of noise. Due to this large variety of different types of noise, we focus on designing an approach that can successfully mitigate any type of the aforementioned artifacts (if possible).

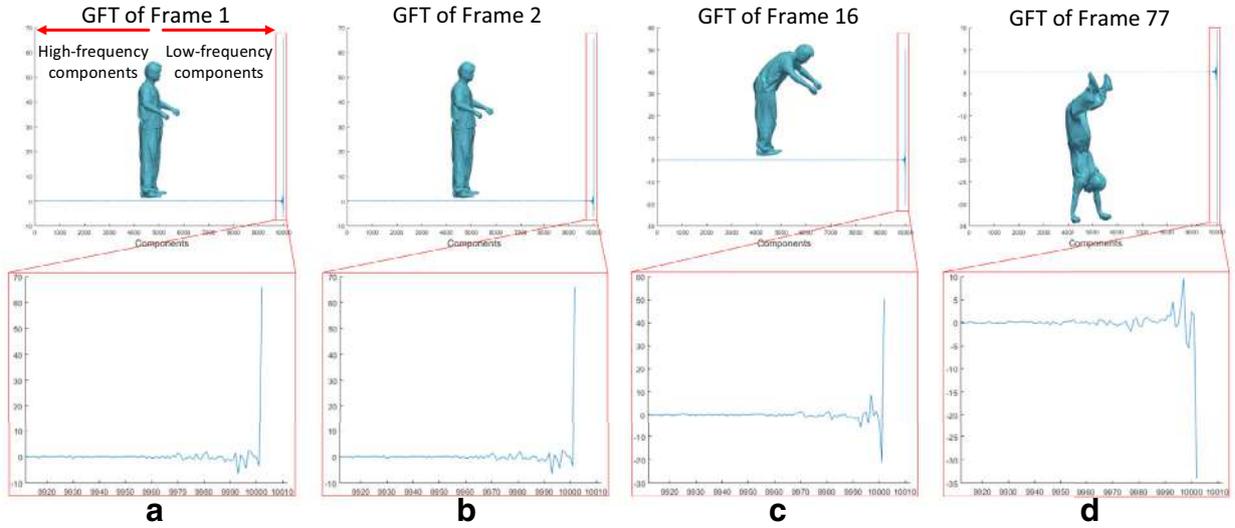
### 3.4. Overview of our method

The proposed method exploits the coherence of the GFT coefficients corresponding to the high spectral frequencies of sequential frames in soft and rigid body animations. On the other hand, the low-frequency components, representing the main shape of a 3D object, have a different form since the shape of the moving object is changing frame by frame. To be more specific, in rigid and soft body animations the 3D shape inevitably changes, affecting the low-frequency components, while the high-frequency components of GFT remain almost unaffected. Fig. 3 presents an example of four different frames of the same dynamic mesh sequence and their corresponding spectral components. As we can see, only a few components (representing the low-frequency components) significantly change while the rest values remain almost the same. The “right” values of the GFT coefficients represent the low-frequency components (i.e. large-scale spatial features) while the “left” values represent the high-frequency components (i.e. small-scale spatial features) (see Fig. 3(a)).

Low-frequency components are essential for the proper representation of a mesh since they are related to the basic shape. On the other hand, changes in the high frequencies are not easily perceived, depending of course of the amount of the components which will be removed. In Fig. 4, we present an example in which different amount of high-frequency components have been removed from a mesh. Specifically, in Fig. 4(e), we can see that even if 90% of the high-frequency components are removed, the basic shape of the 3D object is still recognizable.



**Fig. 2.** (a) Original mesh (armadillo frame 1) and the corresponding noisy meshes affected by: (b) codyyac compression by Vasa and Skala [31], (c) FAMC-DCT compression by Mamou et al. [32], (d) gaussian, (e) impulsive, (f) spatial masking, (g) uniform noise.



**Fig. 3.** GFTs ( $\hat{\mathbf{V}}_x \in \mathbb{R}^{k \times 1}$ ) of different frames of the same dynamic 3D mesh (Handstand model). We can easily observe that most of the components have similar magnitude values while very few components (low-frequency components), representing the information related to shape of the object, have significant different values.

### 3.4.1. Temporal matrices and estimation of the ideal number of the remaining low-frequency components

The main objective of this research is the accurate estimation of the denoised GFTs, in order to use them for the reconstruction of the denoised vertices by applying the IGFT according to Eq. (6). Although the matrix  $\mathbf{U}$  has been estimated once, according to Eq. (4), it is used for the estimation of the  $n$  GFTs  $\hat{\mathbf{V}}_j = [\hat{\mathbf{v}}_{1j}; \hat{\mathbf{v}}_{2j}; \dots; \hat{\mathbf{v}}_{kj}]$  of any other frame  $\forall j = 1, \dots, n$ , according to Eq. (5), where  $\hat{\mathbf{v}}_{ij} = [\hat{v}_{xij} \ \hat{v}_{yij} \ \hat{v}_{zij}] \ \forall i = 1, \dots, k$ . Once the GFTs coefficients have been estimated, we create 3 coherent matrices  $\mathbf{E}_x, \mathbf{E}_y, \mathbf{E}_z$  where  $\mathbf{E}_i \in \mathbb{R}^{n \times k} \ \forall i \in \{x, y, z\}$ , according to:

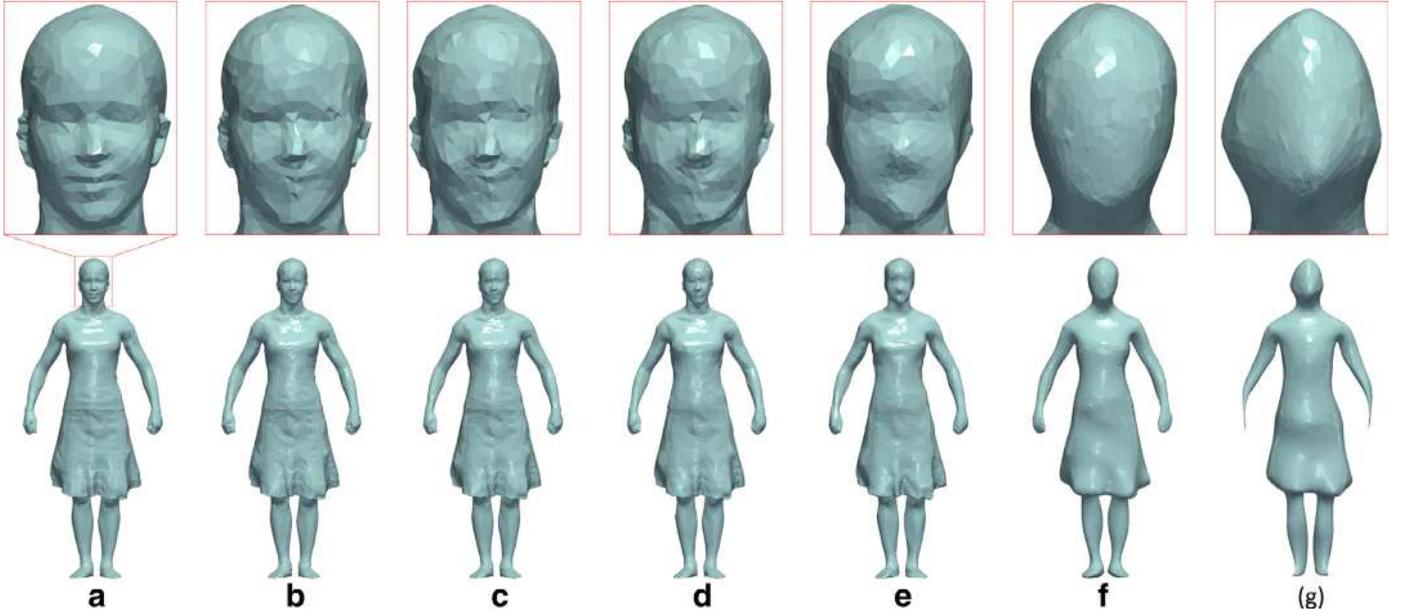
$$\mathbf{E}_i = \begin{bmatrix} \hat{\mathbf{v}}_{i(1:\bar{k})1} \\ \hat{\mathbf{v}}_{i(1:\bar{k})2} \\ \vdots \\ \hat{\mathbf{v}}_{i(1:\bar{k})n} \end{bmatrix} = \begin{bmatrix} \hat{v}_{i11} & \hat{v}_{i21} & \dots & \hat{v}_{i\bar{k}1} \\ \hat{v}_{i12} & \hat{v}_{i22} & \dots & \hat{v}_{i\bar{k}2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{v}_{i1n} & \hat{v}_{i2n} & \dots & \hat{v}_{i\bar{k}n} \end{bmatrix}, \quad \forall i \in \{x, y, z\} \quad (7)$$

where  $\hat{\mathbf{v}}_{i(1:\bar{k})n}$  denotes the  $\bar{k}$  first components of the  $n^{\text{th}}$  frame GFT of the  $i \in \{x, y, z\}$  coordinates. It is worth mentioning that we do not use all the  $k$  components of the GFT for the generation of the

temporal matrices but only the  $\bar{k}$  highest frequency. The main reasons why we exclude a number of low-frequency components are:

- The low-frequency components represent the basic information of the 3D object's form and they should be preserved otherwise the original shape of the 3D object is deformed or significantly deteriorated.
- Additionally, there are not significant coherences between low frequencies of sequential frames, attributed to the fact that small pose changes result in different low spectral frequencies.
- Finally, it has been observed that the distribution of noise, with respect to the GFT domain, mostly affect the high-frequency components since low frequency errors cannot be easily perceived by the human attention system.

For the estimation of the ideal value of  $\bar{k}$ , which represent the number of the high-frequency components per frame that we use for the creation of the coherent matrix, we follow the next steps. Firstly, we estimate the total energy  $E_s$  of the GFT per each frame, taking into account all the  $k$  components, based on the



**Fig. 4.** (a) Original mesh having 10002 components in total, and reconstructed models while components of the GFT have been removed: (b) 6000 components have been removed, (c) 7000 components have been removed, (d) 8000 components have been removed, (e) 9000 components have been removed, (f) 9800 components have been removed, (g) 9900 components have been removed. (Samba model frame 1).

next formula:

$$E_{s_j} = \sum_{i=1}^k \|\hat{v}_i\|^2, \quad \forall j = 1, \dots, n \quad (8)$$

Then, we assume that the 99.99% of this energy must be preserved and the corresponding low-frequency components must remain unaffected. So we start adding high-frequency components, from the component 1 to  $\bar{k}_j$ , until their cumulative energy reaches the 0.01% of the total energy  $E_{s_j}$ :

$$\sum_{i=1}^{\bar{k}_j} \|\hat{v}_i\|^2 = 10^{-4} E_{s_j}, \quad \forall j = 1, \dots, n \quad (9)$$

where  $\bar{k}_j < k$ . The value of  $\bar{k}_j$  may vary from frame to frame, however, each row of the  $\mathbf{E}$  matrix must have equal length. For this reason, the selected value of  $\bar{k}$  is defined as:

$$\bar{k} = \max(\bar{k}_1, \dots, \bar{k}_n) \quad (10)$$

In Fig. 5, we present an example showing how many components are required for preserving: (i) the 99.99% of the GFT's energy (112 lower frequencies components), (ii) the 99.995% of the GFT's energy (1179 lower frequencies components) and (iii) the 99.999% of the GFT's energy for a frame of the Dinosaur model (12255 lower frequencies components). In Fig. 6, we present an example showing how the number of the unchangeable low-frequency components affect the quality of the reconstruction results. For the evaluation, we use different energy thresholds, in a range of [99.5%–99.999%]. A large threshold value (yellow line) means that we maintain many noisy components while small value (red line) means that we may change components that represent the main shape of the mesh and they must have been unaffected.

### 3.4.2. Properties of robust principal component analysis

Once the temporal matrices  $\mathbf{E}_i, \forall i \in \{x, y, z\}$  have been created, we use RPCA in order to estimate their low-rank representation (i.e., reconstructed results). Generally, RPCA has been used in many applications in the area of 3D meshes processing mostly for outliers removal of unorganized point clouds, Arvanitis et al. [35–37].

Our motivation for using RPCA is based on the observation that the high-frequency components of each GFT have a big coherence

with the corresponding components of other frames, with respect to their form and their magnitude. RPCA specializes in finding the low-rank matrix of coherent data (i.e., very relevant data). Additionally, despite the fact that the noise has a unified distribution in the spatial domain (all vertices of the 3D surface can be affected equally), we observed that the noise follows a sparse distribution at the GFT domain in which the high-frequency components are mostly affected. One of the contributions of this approach is that we use RPCA to estimate the low-rank representation (smoothed) of the GFTs values in the spatiotemporal domain which are equal to a denoised representation of the 3D animation in the spatiotemporal domain.

### 3.4.3. Estimation of the low-rank matrix using an RPCA approach

Generally, a coherent matrix  $\mathbf{K}$ , may be decomposed as:

$$\mathbf{K} = \mathbf{S} + \mathbf{N} \quad (11)$$

where  $\mathbf{S}$  is a low-rank matrix representing the real (denoised) data while  $\mathbf{N}$  is a sparse matrix representing the space where the outliers lie. According to Candès et al. [38], the low-rank matrix  $\mathbf{S}$  can be recovered by solving the following convex optimization problem:

$$\text{minimize } \|\mathbf{S}\|_* + \lambda \|\mathbf{N}\|_1, \quad \text{subject to: } \mathbf{S} + \mathbf{N} = \mathbf{K} \quad (12)$$

where  $\|\mathbf{S}\|_*$  denotes the nuclear norm of the matrix which is the sum of the singular values of  $\mathbf{S}$ . This convex problem can be solved using an Augmented Lagrange Multiplier (ALM) algorithm, as described by Lin et al. [39]:

$$l(\mathbf{S}, \mathbf{N}, \mathbf{Y}, \mu) \doteq \|\mathbf{S}\|_* + \lambda \|\mathbf{N}\|_1 + \langle \mathbf{Y}, \mathbf{K} - \mathbf{S} - \mathbf{N} \rangle + \frac{\mu}{2} \|\mathbf{K} - \mathbf{S} - \mathbf{N}\|_F^2 \quad (13)$$

Nevertheless, despite the effectiveness and the very good results, presented also in the work of Lalos et al. [40], the exact decomposition of the Eq. (11) does not always exist especially in real noisy data  $\mathbf{E}$  as those considered in the following section. In this case, an adaptive model is required, taking into account the presence of noise  $\mathbf{E} = \mathbf{K} + \mathbf{G}$ . So the matrix  $\mathbf{E}$  can be decomposed as:

$$\mathbf{E} = \mathbf{S} + \mathbf{N} + \mathbf{G} \quad (14)$$

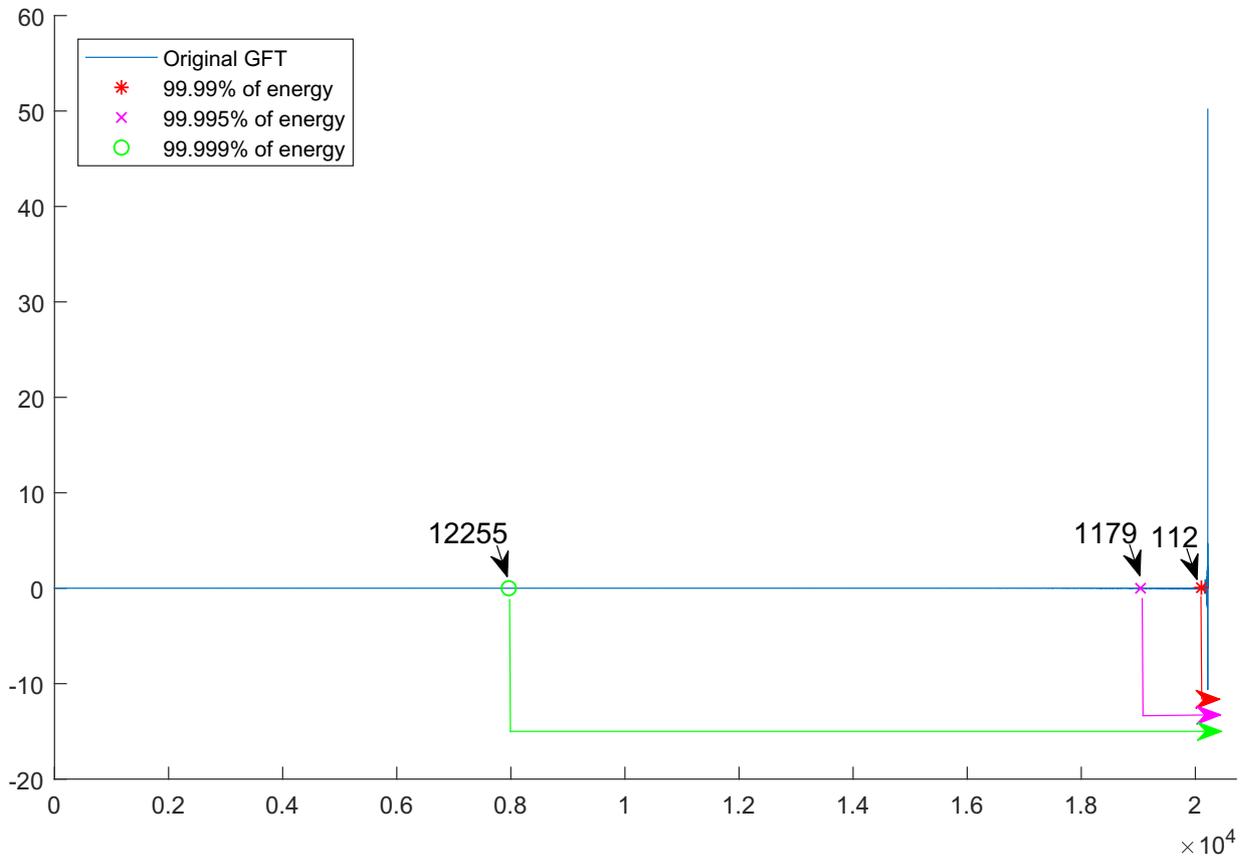


Fig. 5. Energy component (Dinosaur model with 20,218 vertices).

**Bouncing model denoising using different energy remaining approaches**

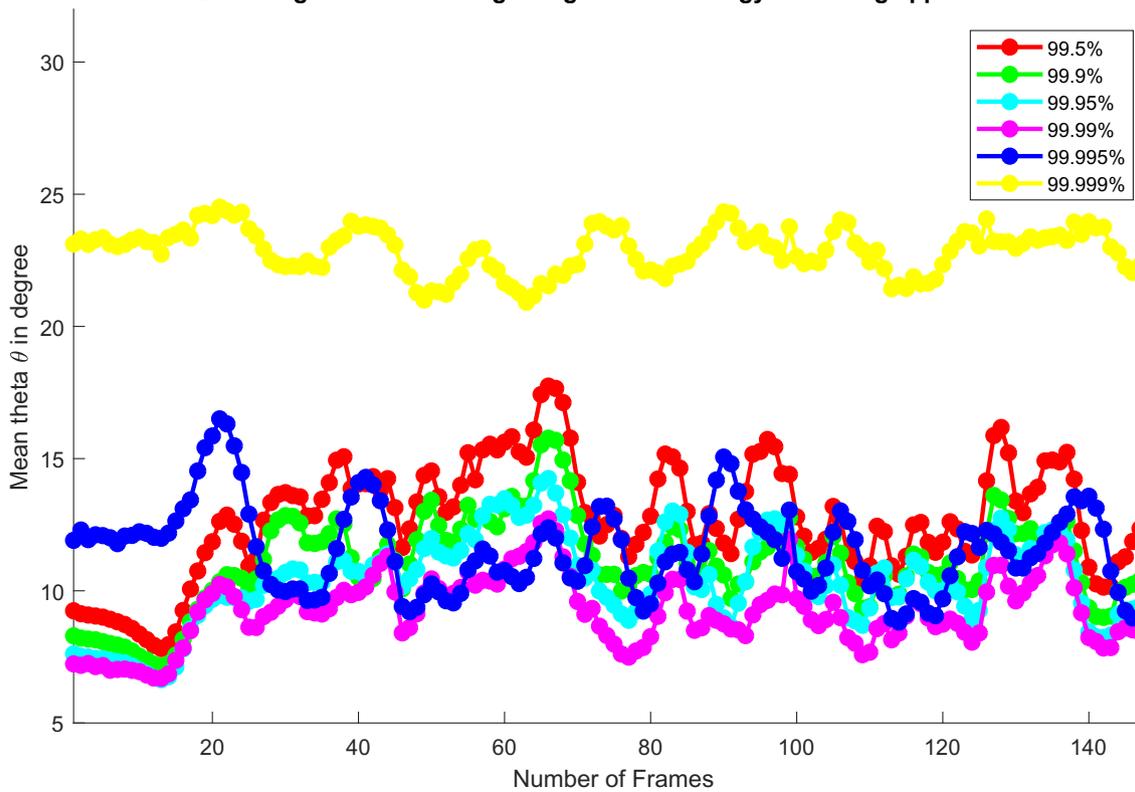


Fig. 6. Bouncing model denoising using different energy remaining approaches. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)

where  $\mathbf{S} + \mathbf{N}$  approximates  $\mathbf{E}$  and  $\mathbf{G}$  is the noisy part. For the sake of simplicity, in this analysis, we use the notation  $\mathbf{E}$  for the coherent matrix, representing however all the cases of Eq. (7). For this kind of problems, Zhou and Tao [41] proposed that the low-rank matrix can be modeled in a bilateral factorization form  $\mathbf{W}\mathbf{T}$  for the purpose of developing an SVD free algorithm. We follow the same line of thought, by replacing  $\mathbf{S}$  with its bilateral factorization:

$$\mathbf{S} = \mathbf{W}\mathbf{T} \quad (15)$$

and regularizing the  $l_1$  norm of the entries of the sparse matrix  $\mathbf{N}$ :

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{T}, \mathbf{N}} \|\mathbf{E} - \mathbf{W}\mathbf{T} - \mathbf{N}\|_F^2 + \lambda \|\text{vec}(\mathbf{N})\|_1 \\ \text{subject to: } \text{rank}(\mathbf{W}) = \text{rank}(\mathbf{T}) \leq r \end{aligned} \quad (16)$$

The  $l_1$  regularization induces soft-thresholding in updating  $\mathbf{N}$ , which is faster than sorting, caused by cardinality constraint, as suggested in a previous work of Zhou and Tao [42] and by Xiong et al. [43]. Optimizing the matrices  $\mathbf{W}$ ,  $\mathbf{T}$  and  $\mathbf{N}$  in Eq. (16), we take the following updating rules:

$$\begin{aligned} \mathbf{W}^{(t+1)} &= (\mathbf{E} - \mathbf{N}^{(t)})\mathbf{T}^{(t)T}(\mathbf{T}^{(t)}\mathbf{T}^{(t)T})^+ \\ \mathbf{T}^{(t+1)} &= (\mathbf{W}^{(t+1)T}\mathbf{W}^{(t+1)})^+ \mathbf{W}^{(t+1)T}(\mathbf{E} - \mathbf{N}^{(t)}) \\ \mathbf{N}^{(t+1)} &= \mathcal{D}_\lambda(\mathbf{E} - \mathbf{W}^{(t+1)}\mathbf{T}^{(t+1)}) \end{aligned} \quad (17)$$

where  $(t)$  denotes the  $t^{\text{th}}$  iteration,  $(\cdot)^+$  is the Moore–Penrose pseudo-inverse and  $\mathcal{D}_\lambda$  is an element-wise soft thresholding operator with threshold  $\lambda$  such that:

$$\mathcal{D}_\lambda = \{\text{sign}(\mathbf{E}_{ij})\max(|\mathbf{E}_{ij}| - \lambda, 0) : (i, j) \in [n] \times [\bar{k}]\} \quad (18)$$

To simplify the updating rules in Eq. (17), we assume that the product  $\mathbf{W}^{(t+1)}\mathbf{T}^{(t+1)}$  equals to the orthogonal projection of  $\mathbf{E}$  onto the column space of  $\mathbf{W}^{(t+1)}$ . According to Eq. (17), the column space of  $\mathbf{W}^{(t+1)}$  can be represented by arbitrary orthonormal basis for the columns of  $(\mathbf{E} - \mathbf{N}^{(t)})\mathbf{T}^{(t)T}$ . It can be computed as  $\mathbf{Q}$  via fast  $\text{qr}(\cdot)$  decomposition:

$$\mathbf{Q}\mathbf{R} = \text{qr}((\mathbf{E} - \mathbf{N}^{(t)})\mathbf{T}^{(t)T}) \quad (19)$$

Then, the product  $\mathbf{W}^{(t+1)}\mathbf{T}^{(t+1)}$  can be equivalently computed as:

$$\mathbf{W}^{(t+1)}\mathbf{T}^{(t+1)} = \mathbf{Q}\mathbf{Q}^T(\mathbf{E} - \mathbf{N}^{(t)}) \quad (20)$$

According to the above analysis, we can observe that the matrices  $\mathbf{W}^{(t+1)}$  and  $\mathbf{T}^{(t+1)}$  in Eq. (17) can be replaced by  $\mathbf{Q}$  and  $\mathbf{Q}^T(\mathbf{E} - \mathbf{N}^{(t)})$  respectively, while the product  $\mathbf{W}^{(t+1)}\mathbf{T}^{(t+1)}$  is kept the same. These replacements change the Eq. (17), providing a faster updating procedure:

$$\begin{aligned} \mathbf{W}^{(t+1)} &= \mathbf{Q}, \quad \mathbf{Q}\mathbf{R} = \text{qr}((\mathbf{E} - \mathbf{N}^{(t)})\mathbf{T}^{(t)T}) \\ \mathbf{T}^{(t+1)} &= \mathbf{Q}^T(\mathbf{E} - \mathbf{N}^{(t)}) \\ \mathbf{N}^{(t+1)} &= \mathcal{D}_\lambda(\mathbf{E} - \mathbf{W}^{(t+1)}\mathbf{T}^{(t+1)}) \end{aligned} \quad (21)$$

### 3.4.4. Reconstruction of the denoised model

The low-rank matrix  $\mathbf{S} \in \mathbb{R}^{n \times \bar{k}}$ , constituting the denoised values, is estimated via the Eq. (15). More specifically, each  $i$  row of this matrix consists of the  $\bar{k}$  high-frequency denoised components of the GFT values of the  $i$  frame. The total denoised GFT matrix  $\hat{\mathbf{V}}_i$  of each  $i$  frame is estimated according to:

$$\hat{\mathbf{V}}_i = [ \underbrace{\mathbf{S}_i[1 : \bar{k}_i]}_{\bar{k}_i \text{ components}} \quad \underbrace{\hat{\mathbf{V}}_i[(k - \bar{k}_i + 1) : k]}_{(k - \bar{k}_i) \text{ components}} ], \quad \forall i = 1, \dots, n \quad (22)$$

keeping the  $\bar{k}_i \leq \bar{k}$  denoised components of the  $i^{\text{th}}$  row of the matrix  $\mathbf{S}$ , following the assumption presented in Section 3.4.1. Then, the final denoised vertices  $\hat{\mathbf{V}}_i$  of each  $i$  frame are estimated using the IGFT, applied to the denoised GFT matrix  $\hat{\mathbf{V}}_i$ :

$$\hat{\mathbf{V}}_i = \mathcal{T}^{-1}(\hat{\mathbf{V}}_i), \quad \forall i = 1, \dots, n \quad (23)$$

As shown in Eq. (23), the reconstruction of the denoised models can be performed either in an adaptive (frame by frame) or in a block adaptive mode (e.g., groups of frames) increasing, even more, the time-efficiently performance of the proposed method. All the aforementioned steps of the proposed approach are summarized at the Algorithm 1.

---

#### Algorithm 1: Spectral Denoising of Dynamic 3D Meshes.

---

**Input** : Noisy sequence of meshes  
 $\tilde{\mathbf{A}} = [\tilde{\mathcal{M}}_1; \tilde{\mathcal{M}}_2; \dots; \tilde{\mathcal{M}}_n];$   
**Output**: Denoised 3D animation  
 $\hat{\mathbf{A}} = [\hat{\mathcal{M}}_1; \hat{\mathcal{M}}_2; \dots; \hat{\mathcal{M}}_n];$

- 1 Decompose the Laplacian matrix  $\mathbf{L}$  of the first frame Eq. (4);
- 2 **for**  $i = 1 \dots n$  **do**
- 3     Estimate the GFT  $\hat{\mathbf{V}}_i$  of the vertices via Eq. (5)
- 4     Estimate the total energy  $E_{s_i}$  of the GFT matrix via Eq. (8)
- 5     Estimate the ideal value  $\bar{k}_i$  of high-frequency components via Eq. (9)
- 6 **end**
- 7 Select the common used value of  $\bar{k}$  via Eq. (10)
- 8 **for**  $\forall j \in \{x, y, z\}$  **do**
- 9     Create the coherent matrix  $\mathbf{E}_j$  via Eq. (7)
- 10    Estimate the low-rank matrix  $\mathbf{S}_j$  via Eqs. (19)–(21)
- 11 **end**
- 12 **for**  $i = 1 \dots n$  **do**
- 13    **for**  $\forall j \in \{x, y, z\}$  **do**
- 14       Estimate the denoised GFT  $\hat{\mathbf{V}}_{ji}$  via Eq. (22)
- 15       Estimate the denoised vertices  $\hat{\mathbf{V}}_{ji}$  of the mesh via Eq. (23)
- 16    **end**
- 17 **end**

---

## 4. Results

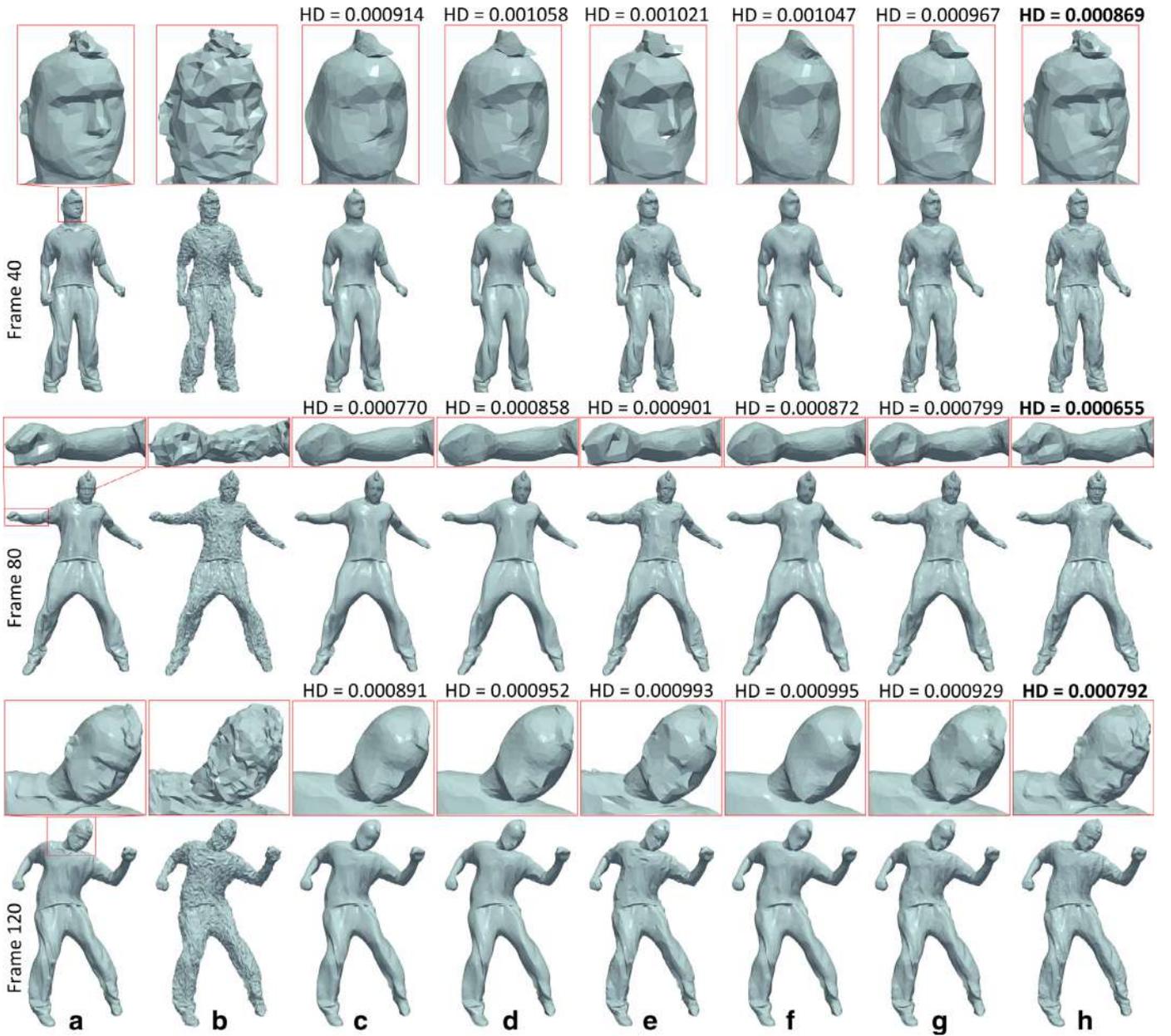
In this section, we present the results of our approach using a variety of dynamic 3D models affected by different types of noise. Extensive evaluation studies carried out using a broad set of complex noise patterns and models with different geometrical features, verify the superiority of our approach as compared to other state-of-the-art methods, in terms of reconstruction quality and computational complexity.

### 4.1. Datasets

The experiments and any other presented figure of this work is carried out using dynamic 3D models of three different well-known datasets.

- A dataset consisting of artificial sequences of moving models with different types of noise by Torkhani et al. [44].
- A dataset consisting of real motion capture animations representing humans (real scanned models) in different kind of motion scenarios by Vlastic et al. [45].
- A dataset consisting of real motion silhouettes captured using joint skeleton tracking by Gall et al. [6].

At this point, it should be noted that the noisy models, affected by Gaussian noise, are created using the dataset of Vlastic et al. [45], by adding noise to the vertices of the ground truth meshes



**Fig. 7.** Three different frames (i.e., 40, 80 and 120) of Bouncing model. (a) Original mesh, (b) Noisy mesh, (c) Bilateral normal denoising by Zheng et al. [16], (d) Fast and effective by Sun et al. [8], (e)  $L_0$  minimization by He and Schaefer [11], (f) Guided mesh normal filtering by Zhang et al. [7], (g) Two stage graph spectral processing by Arvanitis et al. [10], (h) Our approach.

along the vertex normals, similar to Zhang et al. [7]. The intensity of the noise is described using a relative variance parameter:

$$\sigma_l = \frac{\sigma}{\hat{l}} \quad (24)$$

where  $\sigma$  denotes the variance of the Gaussian function, and  $\hat{l}$  is the average edge length  $l$  of the ground truth mesh.

#### 4.2. Metrics

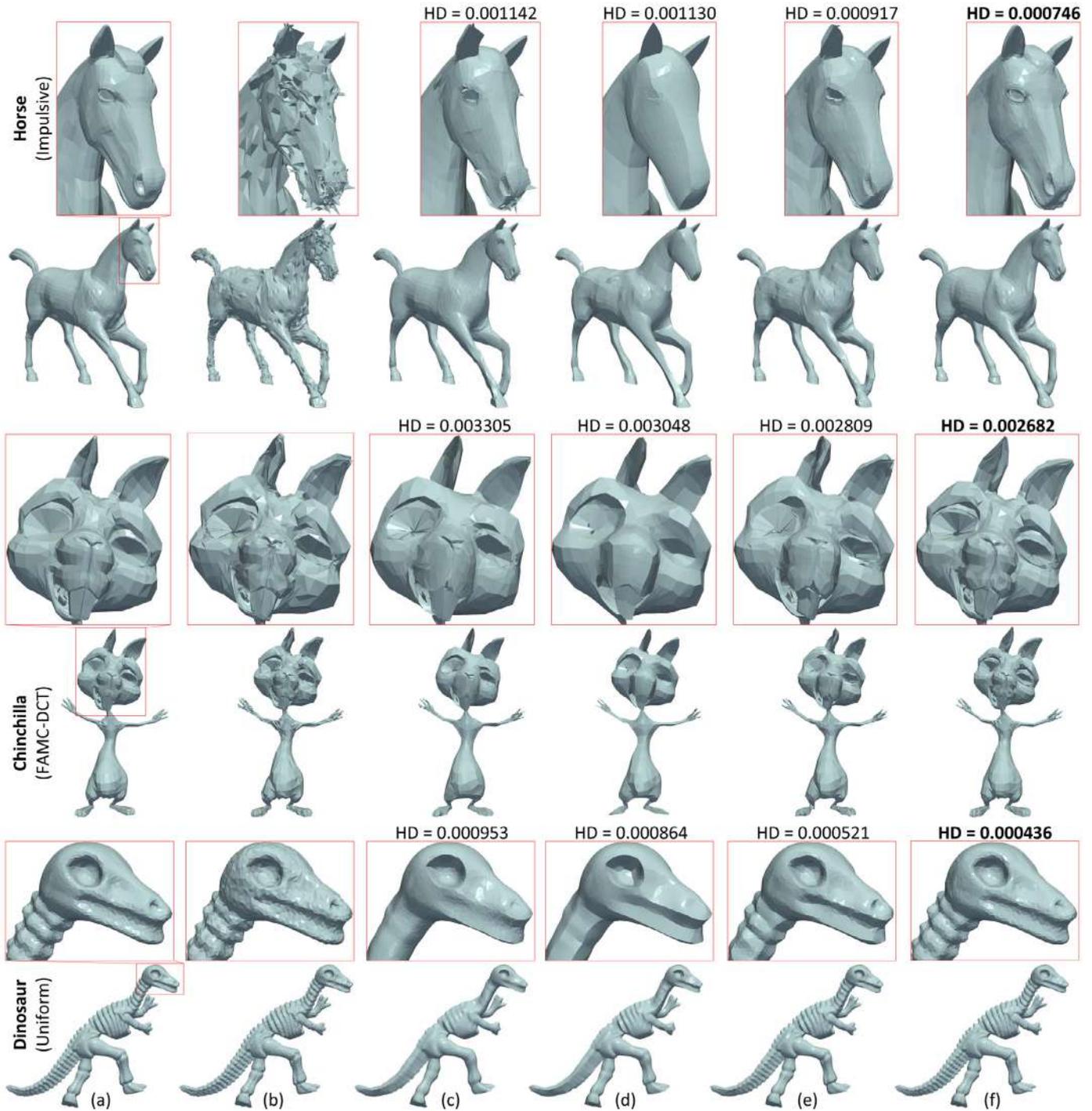
The quality of the reconstructed results is evaluated using a variety of different metrics that are shortly presented below:

- $\theta$ : represents the mean angle  $\gamma$  (expressed in degree) between the normals of the ground truth face and the resulting face normals.

- HD: representing the average one-sided Hausdorff distance (HD) from the denoised mesh to the known ground truth mesh.
- Heatmap visualization which highlights, in different colors (colormap), the angle  $\gamma$  between the normals of the denoised  $\hat{\mathcal{M}}$  and original mesh  $\mathcal{M}$  per each vertex. Dark blue color denotes a big similarity between two normals (practically the angle  $\gamma$  goes to zero), while dark red color denotes a big difference in respect to their directions (please refer to Figs. 9 and 10).

#### 4.3. Experimental results

The quality performance of the proposed technique is evaluated by comparing its denoising results with them of other well-known and robust techniques of the literature, such as: (i) bilateral normal denoising by Zheng et al. [16], (ii) fast and effective by Sun et al.



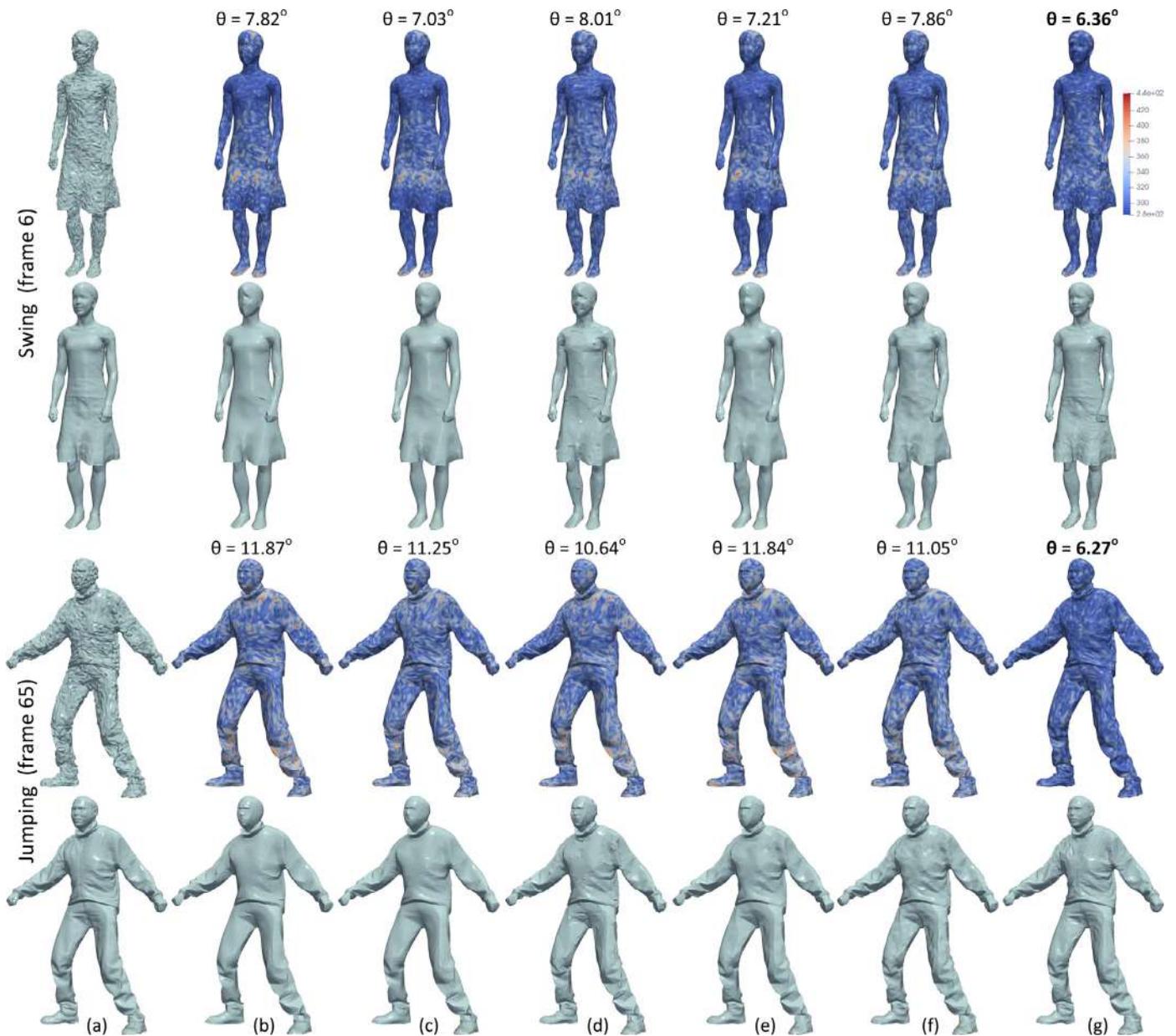
**Fig. 8.** Denoising results of different noisy models (i.e., Horse, Chinchilla, Dinosaur) affected by different types of noise (i.e., Impulsive, Noise because of compression using the FAMC-DCT method and Uniform noise). (a) Original mesh, (b) Noisy mesh, (c) Bilateral normal denoising by Zheng et al. [16], (d) Guided mesh normal filtering by Zhang et al. [7], (e) Two stage graph spectral processing by Arvanitis et al. [10], (f) Our approach.

[8], (iii)  $l_0$  minimization by He and Schaefer [11], (iv) guided mesh normal filtering by Zhang et al. [7] and (v) two stage graph spectral processing by Arvanitis et al. [10].

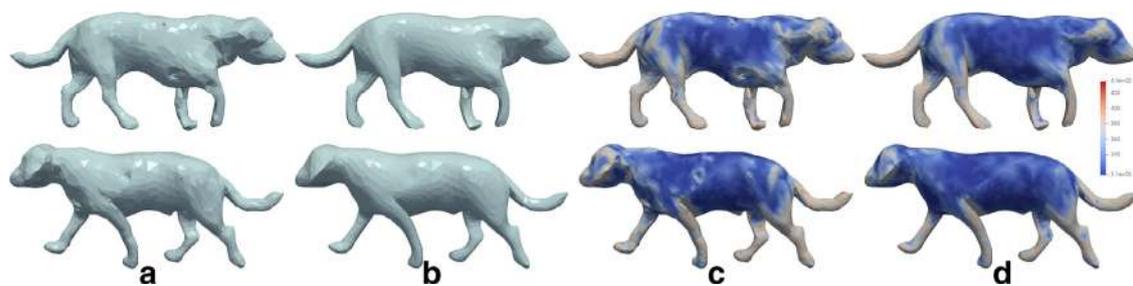
For the experiments, we chose a variety of different models and types of noise. In Table 1, we present information related to the noisy models that are used, namely: (i) the number of vertices, (ii) the number of faces, (iii) the number of frames that each animation has and (iv) the type of noise which each animation has been affected with.

The benefits of our method are apparent in all of the following experimental scenarios. In Fig. 7, we present the denoised results of three different frames (i.e., 40, 80 and 120) of the same model (Bouncing) affected by Gaussian noise. We additionally provide the values of the HD metric, for each reconstructed model, and enlarge details (in red boxes) providing an easier comparison among the meshes and techniques.

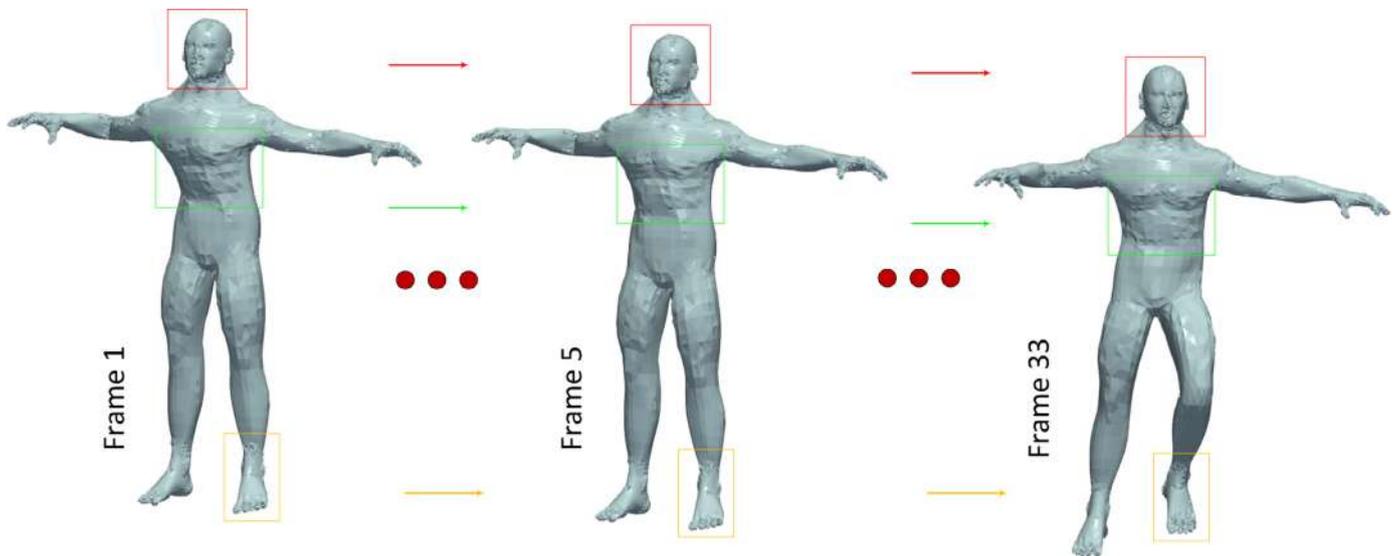
Similar results are presented in Fig. 8. However, in this case, each model has been affected by a different noise patterns. More



**Fig. 9.** (a) Noisy [up] and Original [down] meshes of two models (i.e., Swing, Jumping). We also provide heatmap of metric  $\theta$  [up] and denoising results [down] using: (b) Bilateral normal denoising by Zheng et al. [16], (c) Fast and effective by Sun et al. [8], (d)  $L_0$  minimization by He and Schaefer [11], (e) Guided mesh normal filtering by Zhang et al. [7], (f) Two stage graph spectral processing by Arvanitis et al. [10], (g) Our approach. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)



**Fig. 10.** Two different frames of the Dog animated model [6] and the corresponding heatmap visualization of the mean curvature for the: (a)-(c) Real noisy meshes, (b)-(d) Denoised using our approach. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)



**Fig. 11.** Different frames of the same animated 3D model (Human) having been affected by spacial masks of noise. This means that each frame has been affected by the same magnitude of noise in the same areas.

**Table 1**  
Brief description of the used animated noisy models.

Name of model	Vertices	Faces	Frames	Type of noise
Bouncing	10,002	20,000	150	Gaussian $\sigma_l = 0.2$
Chinchilla	4307	8550	84	FAMC-DCT
Dinosaur	20,218	40,432	152	Uniform
Dog	2502	5000	59	Capturing method [6]
Horse	8431	16,858	48	Impulsive
Human	18,890	37,776	162	Spacial mask
Jumping	9971	19,938	150	Gaussian $\sigma_l = 0.2$
Swing	10,002	20,000	150	Gaussian $\sigma_l = 0.2$

specifically, we show the reconstructed results of three synthetic animations (i.e., Horse, Chinchilla, Dinosaur) affected by (i) impulsive noise<sup>1</sup>, (ii) noise attributed to lossy compression using the FAMC-DCT method<sup>2</sup> and (iii) uniform noise, respectively. This figure highlights the effectiveness of our approach which efficiently reconstructs the denoised models without making special assumptions about the distribution and the type of noise. More importantly, the steps of the proposed method are always the same, as well as the used parameters, making it ideal for using it under different cases without requiring any special parameterization. Our method handles each model with exactly the same way regardless of the geometry of the model (e.g., different geometrical features, intense edges and corners, etc.) or the type of noise.

In Fig. 9, we present the reconstructed results of different techniques, for one frame of two different animated models (namely, the frame 6 of the Swing model and frame 65 of Jumping model). We additionally provide the heatmap visualization of the angle  $\gamma$  colorizing differently each vertex according to the value of  $\gamma$  and the mean angle  $\theta$ . As it can be seen, in any of the presented examples, the proposed method overcome the results of the other comparison methods.

Fig. 10 illustrates the denoising results of our method in models with real noise due to the capturing technique by Gall et al. [6]. Our approach removes the artifacts, appeared in the surface of the models, preserving the small-scale features. Additionally, we

provide the heatmap visualization of the mean curvature for easier comparison.

## 5. Conclusions and open issues

In this work, we presented an approach which performs feature preserving denoising on a sequence of noisy frames that constitute a dynamic 3D mesh. The proposed method achieves plausible reconstruction results using an RPCA approach applied to a coherent matrix  $\mathbf{E}$  consisting of the high frequencies of sequential frames of soft and rigid body animations. More specifically, each row of this matrix consists of the  $\bar{k}$  high-frequency components of the corresponding frame. The low-frequency components are kept unaffected since they represent the basic pose of the mesh. Finally, the denoised vertices occur by applying IGFT to the denoised GFT components.

Nonetheless, despite the very good results that this method provides, there are still limitations that need to be addressed. The extension of the proposed method to dynamic meshes with varying connectivity and a different number of vertices still remains a challenge. Moreover, the performance of the proposed approach is also deteriorated if we assume that the exact same type of noise (in respect of the magnitude and direction) is applied in any sequential frame. Fig. 11 illustrates an example in which corresponding vertices of a sequence of meshes have been affected by spatial masks of noise and as a result, they maintain the exact the same form in any frame.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- [1] Yadav SK, Reitebuch U, Polthier K. Mesh denoising based on normal voting tensor and binary optimization. *IEEE Trans Vis Comput Graph* 2018;24(8):2366–79. doi:10.1109/TVCG.2017.2740384.
- [2] Dybedal J, Aalerud A, Hovland G. Embedded processing and compression of 3d sensor data for large scale industrial environments. *Sensors* 2019;19(3):636. doi:10.3390/s19030636.
- [3] Bogo F, Romero J, Loper M, Black MJ. FAUST: dataset and evaluation for 3D mesh registration. In: *Proceedings IEEE conference on computer vision and pattern recognition (CVPR)*. Piscataway, NJ, USA: IEEE; 2014.

<sup>1</sup> Synthetic but simulates real noise of Lidar point clouds [46]

<sup>2</sup> Real noise because of the encoding/decoding processes

- [4] Xu Z, Zhang Q, Cheng S. Multilevel active registration for kinect human body scans: from low quality to high quality. *Multimedia Syst* 2018;24(3):257–70. doi:10.1007/s00530-017-0541-1.
- [5] Anguelov D, Srinivasan P, Koller D, Thrun S, Rodgers J, Davis J. Scape: shape completion and animation of people. In: *Proceedings of the ACM SIGGRAPH 2005 papers*. In: SIGGRAPH '05. New York, NY, USA: ACM; 2005. p. 408–16. doi:10.1145/1186822.1073207.
- [6] Gall J, Stoll C, de Aguiar E, Theobalt C, Rosenhahn B, Seidel H. Motion capture using joint skeleton tracking and surface estimation. In: *Proceedings of the CVPR*. IEEE Computer Society; 2009. p. 1746–53.
- [7] Zhang W, Deng B, Zhang J, Bouaziz S, Liu L. Guided mesh normal filtering. *Comput Graph Forum* 2015;34(7):23–34. doi:10.1111/cgf.12742.
- [8] Sun X, Rosin PL, Martin RR, Langbein FC. Fast and effective feature-preserving mesh denoising. *IEEE Trans Vis Comput Graph* 2007;13(5):925–38.
- [9] Lu X, Deng Z, Chen W. A robust scheme for feature-preserving mesh denoising. *IEEE Trans Vis Comput Graph* 2016;22(3):1181–94. doi:10.1109/TVCG.2015.2500222.
- [10] Arvanitis G, Lalos AS, Moustakas K, Fakotakis N. Feature preserving mesh denoising based on graph spectral processing. *IEEE Trans Vis Comput Graph* 2019;25(3):1513–27. doi:10.1109/TVCG.2018.2802926.
- [11] He L, Schaefer S. Mesh denoising via l0 minimization. *ACM Trans Graph* 2013;32(4) 64:1–64:8. doi:10.1145/2461912.2461965.
- [12] Yadav SK, Reitebuch U, Polthier K. Robust and high fidelity mesh denoising. *IEEE Trans Vis Comput Graph* 2018 1–1. doi:10.1109/TVCG.2018.2828818.
- [13] Wei M, Huang J, Xie X, Liu L, Wang J, Qin J. Mesh denoising guided by patch normal co-filtering via kernel low-rank recovery. *IEEE Trans Vis Comput Graph* 2018 1–1. doi:10.1109/TVCG.2018.2865363.
- [14] Wang P-S, Liu Y, Tong X. Mesh denoising via cascaded normal regression. *ACM Trans Graph* 2016;35(6) 232:1–232:12. doi:10.1145/2980179.2980232.
- [15] Fleishman S, Drori I, Cohen-Or D. Bilateral mesh denoising. In: *Proceedings of the ACM SIGGRAPH 2003 papers*. In: SIGGRAPH '03. New York, NY, USA: ACM; 2003. p. 950–3. doi:10.1145/1201775.882368. ISBN 1-58113-709-5.
- [16] Zheng Y, Fu H, Au OK-C, Tai C-L. Bilateral normal filtering for mesh denoising. *IEEE Trans Vis Comput Graph* 2011;17(10):1521–30.
- [17] Jones TR, Durand F, Desbrum M. Non-iterative, feature-preserving mesh smoothing. *ACM Trans Graph* 2003;22(3):943–9.
- [18] Wang J, Zhang X, Yu Z. A cascaded approach for feature-preserving surface mesh denoising. *Comput Aided Des* 2012;44(7):597–610. doi:10.1016/j.cad.2012.03.001. <http://www.sciencedirect.com/science/article/pii/S0010448512000516>
- [19] Centin M, Signoroni A. Mesh denoising with (geo)metric fidelity. *IEEE Trans Vis Comput Graph* 2018;24(8):2380–96. doi:10.1109/TVCG.2017.2731771.
- [20] Zeng J, Cheung G, Ng M, Pang J, Yang C. 3d point cloud denoising using graph Laplacian regularization of a low dimensional manifold model; 2018. <https://arxiv.org/abs/1803.07252>
- [21] Dinesh C, Cheung G, Bajic IV, Yang C. Fast 3d point cloud denoising via bipartite graph approximation & total variation; 2018. <https://arxiv.org/abs/1804.10831>
- [22] Wei M, Liang L, Pang W, Wang J, Li W, Wu H. Tensor voting guided mesh denoising. *IEEE Trans Autom Sci Eng* 2017;14(2):931–45. doi:10.1109/TASE.2016.2553449.
- [23] Taubin G. A signal processing approach to fair surface design. In: *Proceedings of the 22nd annual conference on computer graphics and interactive techniques*. In: SIGGRAPH '95. New York, NY, USA: ACM; 1995. p. 351–8. doi:10.1145/218380.218473. ISBN 0-89791-701-4.
- [24] Rustamov RM. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In: *Proceedings of the fifth eurographics symposium on geometry processing*. In: SGP '07. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association; 2007. p. 225–33. ISBN 978-3-905673-46-3; <http://dl.acm.org/citation.cfm?id=1281991.1282022>
- [25] Zhang H, Kaick Ov, Dyer R. Spectral methods for mesh processing and analysis. In: Schmalstieg D, Bittner J, editors. *Eurographics 2007 - state of the art reports*. The Eurographics Association; 2007. doi: 10.2312/egst.20071052.
- [26] Mattei E, Castrodad A. Point cloud denoising via moving rpca. *Comput Graph Forum* 2017;36(8):123–37. doi:10.1111/cgf.13068.
- [27] Pauly M, Gross M. Spectral processing of point-sampled geometry. In: *Proceedings of the 28th annual conference on computer graphics and interactive techniques*. In: SIGGRAPH '01. New York, NY, USA: ACM; 2001. p. 379–86. doi:10.1145/383259.383301. ISBN 1-58113-374-X.
- [28] Rosman G, Dubrovina A, Kimmel R. Patch-collaborative spectral point-cloud denoising. *Comput Graph Forum* 2013. doi:10.1111/cgf.12139.
- [29] Schall O, Belyaev A, Seidel H-P. Adaptive feature-preserving non-local denoising of static and time-varying range data. *Comput Aided Des* 2008;40(6):701–7. doi:10.1016/j.cad.2008.01.011. *Selected Papers from the ACM Solid and Physical Modeling Symposium 2007*; <http://www.sciencedirect.com/science/article/pii/S0010448508000353>
- [30] Remil O, Xie Q, Xie X, Xu K, Wang J. Surface reconstruction with data-driven exemplar priors; 2017. <https://arxiv.org/abs/1701.03230>
- [31] Vasal L, Skala V. CODDYAC: connectivity driven dynamic mesh compression. In: *Proceedings of the 3DTV conference*; 2007. p. 1–4. doi:10.1109/3DTV.2007.4379408.
- [32] Mamou K, Zaharia T, Preteux F. FAMC: The MPEG-4 standard for animated mesh compression. In: *Proceedings of the 15th IEEE international conference on image processing*; 2008. p. 2676–9. doi:10.1109/ICIP.2008.4712345.
- [33] Vallet B, Lévy B. Spectral geometry processing with manifold harmonics. *Comput Graph Forum* 2008;27(2):251–60. doi:10.1111/j.1467-8659.2008.01122.x.
- [34] Lalos AS, Arvanitis G, Dimas A, Moustakas K. Block based spectral processing of dense 3d meshes using orthogonal iterations. *Proceedings of the 13th international joint conference on computer vision, imaging and computer graphics theory and applications: GRAPP, INSTICC, Volume 1*. SciTePress; 2018. p. 122–32. doi:10.5220/0006611401220132. ISBN 978-989-758-287-5.
- [35] Arvanitis G, Spathis-Papadiotis A, Lalos AS, Moustakas K, Fakotakis N. Outliers removal and consolidation of dynamic point cloud. In: *Proceedings of the 25th IEEE international conference on image processing (ICIP)*; 2018. p. 3888–92. doi:10.1109/ICIP.2018.8451099.
- [36] Arvanitis G, Lalos AS, Moustakas K, Fakotakis N. Real-time removing of outliers and noise in 3d point clouds applied in robotic applications. In: Ronzhin A, Rigoll G, Meshcheryakov R, editors. *Interactive collaborative robotics*. Cham: Springer International Publishing; 2017. p. 11–19. ISBN 978-3-319-66471-2.
- [37] Arvanitis G, Lalos AS, Moustakas K, Fakotakis N. Outliers removal of highly dense and unorganized point clouds acquired by laser scanners in urban environments. In: *Proceedings of the international conference on Cyberworlds (CW)*; 2018. p. 415–18. doi:10.1109/CW.2018.00080.
- [38] Candès EJ, Li X, Ma Y, Wright J. Robust principal component analysis? *J ACM* 2011;58(3) 11:1–11:37. doi:10.1145/1970392.1970395.
- [39] Lin Z, Chen M, Ma Y. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *CoRR* 2009;abs/1009.5055.
- [40] Lalos AS, Vlachos E, Arvanitis G, Moustakas K, Berberidis K. Signal processing on static and dynamic 3d meshes: Sparse representations and applications. *IEEE Access* 2019;7:15779–803. doi:10.1109/ACCESS.2019.2894533.
- [41] Zhou T, Tao D. Greedy bilateral sketch, completion & smoothing. In: *Proceedings of the international conference on artificial intelligence and statistics*; 2013.
- [42] Zhou T, Tao D. GoDec: randomized low-rank & sparse matrix decomposition in noisy case. In: *Proceedings of the 28th international conference on international conference on machine learning*. In: ICML'11. USA: Omnipress; 2011. p. 33–40. ISBN 978-1-4503-0619-5; <http://dl.acm.org/citation.cfm?id=3104482.3104487>
- [43] Xiong L, Chen X, Schneider JG. Direct robust matrix factorization for anomaly detection. *Proceedings of the IEEE 11th international conference on data mining*; 2011. p. 844–53.
- [44] Torkhani F, Wang K, Chassery J-M. Perceptual quality assessment of 3D dynamic meshes: Subjective and objective studies. *Signal Process Image Commun* 2015;31(2):185–204. doi:10.1016/j.image.2014.12.008. <https://hal.archives-ouvertes.fr/hal-01118984>
- [45] Vlastic D, Baran I, Matusik W, Popović J. Articulated mesh animation from multi-view silhouettes. *ACM Trans Graph* 2008;27(3):97:1–97:9.
- [46] Awadallah MST. Image analysis techniques for lidar point cloud segmentation and surface estimation, Virginia Tech: Virginia Polytechnic Institute and State University; 2016. Ph.D. thesis.