IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 21, 2018, accepted January 13, 2019, date of publication January 23, 2019, date of current version February 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2894533

Signal Processing on Static and Dynamic 3D Meshes: Sparse Representations and Applications

ARIS S. LALOS[1,2], (Member, IEEE), EVANGELOS VLACHOS[3], GERASIMOS ARVANITIS[2], KONSTANTINOS MOUSTAKAS[2], (Senior Member, IEEE), AND KOSTAS BERBERIDIS[4], (Senior Member, IEEE)

[1]Industrial Systems Institute, ATHENA Research and Innovation Center, 265 04 Platani, Greece
[2]Department of Electrical and Computer Engineering, University of Patras, 265 04 Patras, Greece
[3]Institute for Digital Communications, The University of Edinburgh, Edinburgh EH8 9YL, U.K.
[4]Department of Computer Engineering and Informatics, University of Patras, 265 04 Patras, Greece

Corresponding author: Aris S. Lalos (lalos@isi.gr)

ABSTRACT Nowadays, real-time 3D scanning and reconstruction becomes a requirement for a variety of interactive applications in various fields, including heritage science, gaming, engineering, landscape topography, and medicine. From the introduction of 3D scanning, which allowed the representation of real world or synthetic objects into the virtual world, hardware and software advances have seen tremendous progress. However, despite the continuous improvement of the new generation image sensors and acquisition techniques, the acquired data are often corrupted by the low-frequency noise, outliers, misalignment, missing data, and variations in point density. These effects are amplified if the low-cost sensors and hardware are being used (e.g., mobile devices); thus, the acquisition and communication cost per datum is driven to a minimum. This paper provides a comprehensive review of the ongoing efforts in geometry and signal processing, describing several models from a wide range of signal processing relevant tasks, such as robust principal component analysis, compressive sampling, and matrix completion. Various scalable architectures and optimization algorithms are analyzed and reviewed, revealing significant insights into the fundamental processing operations and the involved implementation tradeoffs. Moreover, the impact of sparse modeling and optimization tools to several 3D mesh processing tasks, such as completion of missing data, feature preserving noise removal, and rejection of outliers, is illustrated via test cases with several constraints posed by the arbitrarily complex animated scenarios. Finally, the identified limitations together with the potential open research directions are also presented for future research efforts toward modeling and optimization for static and dynamic 3D models.

INDEX TERMS Signal processing on static and dynamic meshes, sparse representation theory & algorithms, 3D geometry acquisition and processing.

I. INTRODUCTION

The rapid advancements in computational methods, optics, and graphics computing have significantly contributed to the evolution of 3D scanning technologies and the acquisition of dynamically deforming 3D shapes at sustained video rates. Today's 3D scanning devices are capable of capturing dynamic 3D scenes (e.g., humans in motion, 3D scenes from a moving camera) in real time facilitating several tasks in diverse fields such as archaeology, landscape topography, gaming, engineering and medicine. Virtual and Augmented Reality (VR/AR) which represents a real-time fusion between real-world geometry and 3D graphics being rendered live to the user; immersive technologies to provide immediate feedback to users during 3D scanning; autonomous guidance for robots which permits the rapid respond to changes of their environment (e.g., robotic surgery).

From the early days of photography, there are several attempts to capture shape by optical means. In the 1860s, Francois Villeme invented a process known as photosculpture [1] by using a set of 24 cameras. However, the process

required a substantial investment in cameras, projection and reproduction systems, while a professional sculptor was still needed for fine tuning the pieces. The reliable generation of the shape of real-world scenes has regained substantial interest with the advent of computers. The use of calibrated digital cameras to recover with high accuracy the position of features in the scene is a well-established technique that operates under ambient illumination and is known as *passive approach*. Passive systems offer high accuracy on well-defined features, e.g., corners and edges. However, the output geometry of unmarked surfaces is highly affected by the ambient light since are hard to be measured using passive systems. *Active systems* on the other hand, provide their own illumination and the features to be measured so that they can easily capture surfaces in most environments [2]. There are many *active* 3D scanning technologies while some of the most widely adopted are: (i) structured light systems and (ii) Laser Imaging Detection and Ranging (LIDAR) systems.

The resolution and accuracy of the new generation 3D scanning systems are constantly improving, however, the generated surface usually contain a variety of imperfections (e.g., noise, outliers, sampling density, misalignment and missing data [3]–[5]), owing to occlusions and the scanners physical limitations. These effects pose significant challenges for reconstruction of static and dynamic surfaces representing the rigid body and highly deformable motions. While they can be mitigated by applying effective denoising, in painting, filtering and compression techniques to the dense captured geometry, the tremendous computing resources required due to the high spatiotemporal resolution bring significant challenges to be solved. The challenges become even more demanding in real-time scenarios, such as immersive communications (e.g., streaming) and content creation applications, where decentralized processing in parallelized multicore systems is preferred.

Usually, in order to represent 3D models in applications, polygon modeling via 3D meshes is being used to approximate surfaces. In general, 3D meshes consist of vertices that provide the geometry information and polygons that connect the vertices. Dynamic meshes are defined as a series of static, mainly triangular, meshes representing a 3D animation that usually appears in two forms: *rigid* and *soft-body* motions. In rigid-body animations, the relative position of two adjacent vertices between consecutive 3D meshes stays fixed and the body moves as one entity. Although the assumption that the bodies are rigid simplifies their representation, it does not allow the capturing of many realistic motions of deformable objects. To overcome this limitation, soft-body dynamics has been widely adopted for simulating the motion and properties of deformable objects, also known as soft-bodies. Unlike rigid-bodies, the shape of soft-bodies changes over time, meaning that the relative distance of two vertices between consecutive meshes does not remain fixed. In both cases, the captured geometry has a rich spatiotemporal structure which that can be efficiently exploited using fast and effective approaches to overcome the aforementioned challenges.

Among our motivations for writing this review article were the results of an extensive literature overview, we performed regarding recent advances for the research areas under consideration (mostly for the time window 2006 to 2018). More specifically, the top world's leading scientific, research and development publication indexing databases, Elsevier's Scopus and Web of Science (WoS) by Clarivate Analytics, yield thousand results if keywords as ''low rank'', ''sparsity'', ''3D scanning'' and ''point cloud consolidation'' are included in the searching title or keyword. By inspecting this results, it can be easily observed that the total number of relevant publications has been remarkably increased. More than 100 selected articles, which are closely related to the application of sparse modeling and optimization tools on static and dynamic 3D meshes, have been studied. Motivated by this extensive literature study, the scope of this article is to provide a comprehensive review of recent approaches capitalizing on the low-rank property of the captured geometry of both rigid and soft-body motions and the sparsity of: i) the outliers and ii) the sharp geometric features (e.g, corners and edges), allowing benefits from rank minimization, compressive sampling, and robust principal component analysis when processing the captured geometry of dynamic meshes that might contain outliers, noise, and nonuniformities. This article analyzes and assesses scalable architectures, optimization algorithms and it suggests possible future research directions to be tackled by scientists and engineers in the field [6]. Evaluation studies carried out using captured static and dynamic 3D scenes, reveal fundamental insights into the various signal processing tasks involved when dealing with scanning imperfection attributed to occlusions and physical limitations of the device. Undoubtedly, the presented state of the art computationally intelligent approaches, based on signal processing (SP) methodologies, are expected to play a crucial role in this exciting endeavor and inspire also new SP techniques in the field of static and dynamic 3D mesh processing.

Organization: Basic definitions related to static and dynamic 3D models are provided in Section II. Section III provides a comprehensive review of prior art on exploiting the rich spatiotemporal structure, of the captured static and dynamic 3D geometry, using sparse modeling and optimization tools. Section IV analyzes state of the art centralized optimization algorithms for dealing with a variety of scanning imperfections and the article is wrapped up with a few open research directions in Section VI.

Notation: $\mathbf{A}^T$, $\mathbf{A}^*$, $\mathbf{A}^H$, and $\|\mathbf{A}\|_F$ represent $\mathbf{A}$'s transpose, conjugate transpose, Hermitian transpose, and Frobenius norm; the inner product of two matrices $\mathbf{X}, \mathbf{Y}$ is defined as $\langle \mathbf{X}, \mathbf{Y} \rangle = \text{tr}(\mathbf{X}^H \mathbf{Y})$; diag($\mathbf{x}$) denotes the diagonal matrix which is constructed based on the vector $\mathbf{x}$; $(x)_+ = \max(0, x)$ which represents the positive part of x; $\circ$ denotes the Hadamard (element-wise) product; $[X]_{i,j}$ denotes the element of the matrix $\mathbf{X}$ at the i-th row and j-th column [7]; $\|\mathbf{X}\|_* = \sum_k \sigma_k(\mathbf{X})$ denotes the nuclear norm of the matrix $\mathbf{X}$, where $\sigma_k(\mathbf{X})$ is the k-th singular value of the matrix;

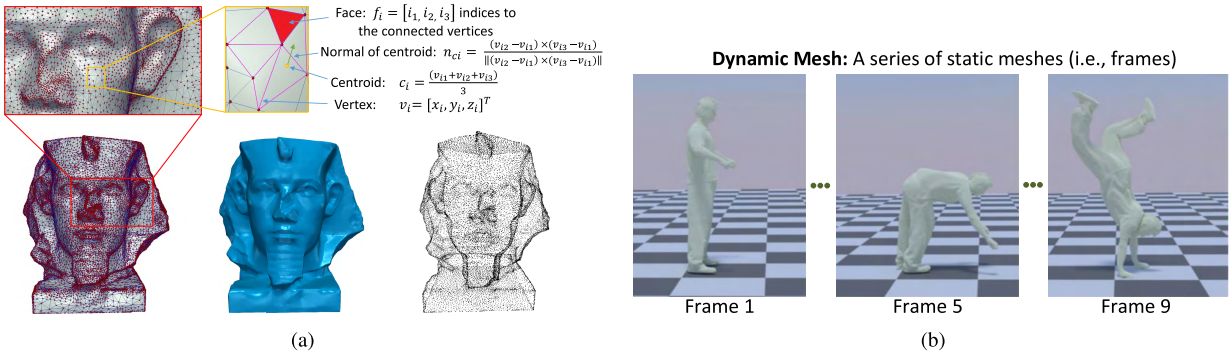


FIGURE 1. Static and Dynamic 3D Models: Definitions. (a) Static 3D mesh and respective point cloud. (b) Dynamic 3D mesh.

Operands $\circ$ and $\otimes$ denote the matrix Hadamard and Kronecker products, respectively the l_p norm of a vector is defined as $\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$ and the l_1 norm of a matrix is defined as the sum of the absolute values of its elements $\|\mathbf{M}\|_{l_1} = \sum_{ij} |M_{ij}|$; $\Re$ represents the set of real numbers.

II. 3D MESHES: BASIC DEFINITIONS AND TOOLS

The analysis and processing of static and highly deformable 3D objects facilitate new discoveries in various scientific areas, such as neuroscience, astrophysics, mechanical and civil engineering [8]. These objects are digitized using different 3D acquisition technologies, including computational tomography (CT), 3D LIDAR, ultrasound, magnetic resonance imaging, and structured light, to name a few. Recently, there has been increasing interest for reliable generation of 3D content in other fields industries, including remote science, entertainment, urban modeling, and heritage science. Digital geometry processing of complex 3D objects lies between acquisition and production and is a field of computer science focusing on the mathematical models and algorithms for processing and analyzing geometric objects, usually represented via polygon models.

In the following subsections, first, we introduce some basic definitions on the 3D static and dynamic meshes and their mathematical representation. Then, we describe vertex and normal filtering operations and the principal component analysis based compression, two standard tools that are used in mesh processing for removing imperfections and/or compressing static and dynamic 3D meshes.

A. BASIC DEFINITIONS

3D meshes are described by a set of vertices corresponding to the geometric information and a set of polygons, also known as faces that determine elementary surfaces, as shown in Fig. 1 (a). The output of a 3D scanning process is a sequence of unstructured point clouds (e.g., vertices), which are then transformed into a surface mesh, after applying a surface reconstruction method that generates the connectivity information. In this work, we focus on triangle meshes since they are the most widely adopted polygon meshes. A triangle mesh $\mathbf{M}$ with n vertices is defined as $\mathcal{M} = (\mathcal{V}, \mathcal{F})$, where $\mathcal{V}$ corresponds to the vertices that represent also a Point Cloud (PC) and $\mathcal{F}$ is a set with the indexed faces. The corresponding set of edges $\mathcal{E}$, are directly derived from sets $\mathcal{V}$ and $\mathcal{F}$. A vertex is defined by its absolute Cartesian coordinates and is represented as a 3×1 vector $\mathbf{v}_i = [x_i,\ y_i,\ z_i]^T$. Therefore the mesh vertices can be also seen as a three-dimensional graph signal $\mathbf{M} = [\mathbf{v}_1,\ \mathbf{v}_2,\ \cdots, \mathbf{v}_n] \in \Re^{3\times n}$. A neighbourhood of $\mathbf{v}_i$ in $\mathbf{M}$ corresponds to the set $\mathcal{N}_i$ of vertex indices j connected to i by an edge (i, j),

$$\mathcal{N}_i = \{j \,|(i, j) \in \mathcal{E}\}. \tag{1}$$

Any vertex that belongs to the neighbourhood $\mathcal{N}_i$ is a topological neighbor of i. The differential (or $\boldsymbol{\delta}$) coordinates of a mesh are calculated as the difference between the coordinates of each vertex $\mathbf{v}_i$ and the coordinates of the neighborhood barycenter, e.g.,

$$\boldsymbol{\delta}_i = \left[\delta_{x_i},\ \delta_{y_i},\ \delta_{z_i}\right]^T = \mathbf{v}_i - 1/d_i \sum_{j\in\mathcal{N}_i} \mathbf{v}_j, \tag{2}$$

where $d_i = |\mathcal{N}_i|$ corresponds to the number of neighbours of vertex i, also known as degree of $\mathbf{v}_i$ [9]. The direction of $\boldsymbol{\delta}_i$ approximates the curvature flow normals as it is also shown in Fig. 2, i.e., $\boldsymbol{\delta}_i = -\mathcal{H}(\mathbf{v}_i)\,\mathbf{n}_i$ [10], where $\mathcal{H}(\mathbf{v}_i)$ represents the mean curvature at $\mathbf{v}_i$ and γ is a closed surface curve around $\mathbf{v}_i$ and $|\gamma|$ is the length of γ. Moreover, it should be noted that for a face $f_i \in \mathcal{F}$, the normal can be computed as:

$$\mathbf{n}_i = \frac{\left(\mathbf{v}_{i_2} - \mathbf{v}_{i_1}\right) \times \left(\mathbf{v}_{i_3} - \mathbf{v}_{i_1}\right)}{\left\|\left(\mathbf{v}_{i_2} - \mathbf{v}_{i_1}\right) \times \left(\mathbf{v}_{i_3} - \mathbf{v}_{i_1}\right)\right\|_2} \tag{3}$$

where $\times$ corresponds to the cross product operation, $\mathbf{v}_{i_1}$, $\mathbf{v}_{i_2}$ and $\mathbf{v}_{i_3}$ are the position of the associated vertices in a fixed orientation. Each normal $\mathbf{n}_i$ is associated with the face centroid $\mathbf{c}_i = \left(\mathbf{v}_{i_1} + \mathbf{v}_{i_2} + \mathbf{v}_{i_3}\right)/3$ as shown in Fig. 3.

B. FILTERING MESH GEOMETRY

Mesh filtering is usually adopted for improving imperfect meshes obtained by 3D acquisition devices (e.g., surface reconstruction from CT scans). Taubin [10] first introduced the use of Laplacian operators for discrete geometry processing. To overcome the smoothing effects, several

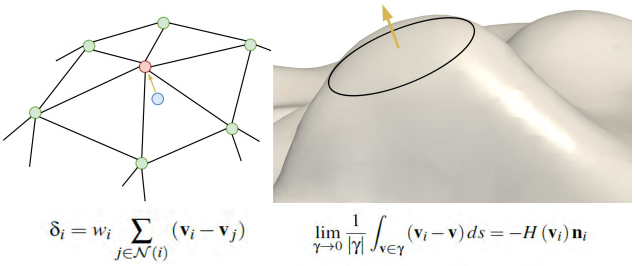


FIGURE 2. The direction of the differential coordinate vector approximates the local normal direction [10].

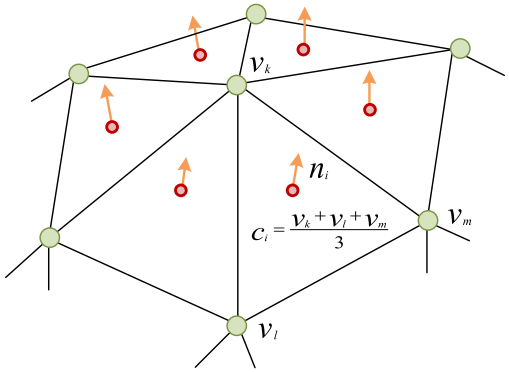


FIGURE 3. Normal vectors $\mathbf{n}_i$ associated with the face centroids $\mathbf{c}_i$.

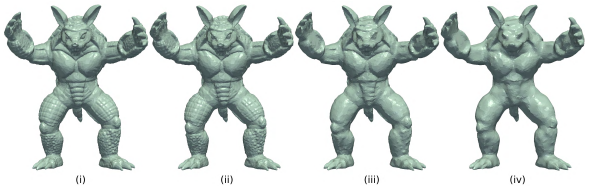


FIGURE 4. Compact representation of the Armadillo geometry using a different number of eigenvectors of the graph Laplacian: (i) original geometry, (ii) compact representation using 90% (those that correspond to small eigenvalues) of the eigenvectors (iii) compact representation using 25% of the eigenvectors (iv) compact representation using 10% of the eigenvectors. Note that the surface attains more high-frequency detail as we increase the number of eigenvectors.

works suggested the use of the bilateral filter [11], either directly to the vertices [12], [13] or to the normals associated with the faces. In the second case, the feature preserving bilateral process is then followed by a vertex updating scheme [14]–[16]. In the rest part, we focus on approaches that have been widely used for filtering either the vertex positions and/or the normals associated with the centroids of the faces.

1) FILTERING VERTEX POSITIONS

Several works make use of the Laplacian operator for evaluating the delta coordinates and filtering the vertex coordinates. Assuming that the connectivity information is provided (e.g., set of faces $\mathcal{F}$), then we can define the adjacency matrix $\mathbf{C} \in \Re^{n \times n}$ with elements:

$$\mathbf{C}_{(i,j)} = \begin{cases} w_{ij} & (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where w_{ij} can either take the value 1 (binary adjacency) or weighted values (e.g., $e^{(-\|\mathbf{v}_i - \mathbf{v}_j\|_2^2)/(2\sigma^2)}$) in case of weighted adjacency, with σ denoting the variance of the threshold for edge existence. The Laplacian operator can be written also in matrix form as:

$$\mathbf{L} = \mathbf{D} - \mathbf{C} \tag{5}$$

and $\mathbf{D}$ is the diagonal degree matrix with $\mathbf{D}_{(i,i)} = |N(i)|$. In the literature, the normalized Laplacian is also known as graph Laplacian and is defined as $\bar{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. Matrix $\bar{\mathbf{L}}$ has the same spectrum with the symmetric *Tutte Laplacian* [17] $\bar{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{D}^{-1}\mathbf{L}$, symmetry facilitates the computation of the eigenvectors of $\mathbf{L}$.

The Laplacian operator can be also defined if we assume that the connectivity information is not available. In such case, we usually consider the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ where $\mathcal{W} = \{w_{ij} \mid i,j = 1, \ldots, n\}$ is the set with the non-negative weights and $\mathcal{E}$ is the set of edges. These graph weights can be constructed either by utilizing the $\epsilon \in \Re$ neighborhoods (ϵ-N) and/or the $\kappa \in \mathbb{N}$ nearest neighbors (κ-NN). At this point it should be noted that the ϵ-N graphs are symmetric and more geometrically meaningful. However ϵ should be carefully selected, since there are cases where specific values of ϵ could lead to dense or disconnected graphs, contrary to the κ-NN graphs which usually correspond to connected graphs. In the following part we employ κ-NN binary weighted graphs for the derivations of the Laplacian matrix $\mathbf{L}$ which is defined in Eq. (5) and can be employed in order to provide a proximity metric for the point cloud geometry. Let as assume that the eigenvalue decomposition of the constructed graph Laplacian $\mathbf{L}$ is written as:

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{6}$$

where $\mathbf{\Lambda}$ is a diagonal matrix consisting of the eigenvalues of $\mathbf{L}$ and $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ is the matrix with the eigenvectors $\mathbf{u}_i \in \Re^{n \times 1}\ \forall\ i = 1, \ldots, n$ that span the graph Fourier basis. Similar to classical Fourier transform, the eigenvectors and eigenvalues of the Laplacian matrix $\mathbf{L}$ provide a spectral interpretation of the 3D signal [18]. The Graph Fourier Transform (GFT) and the inverse GFT of the vertex coordinates is defined as:

$$\text{GFT} : \bar{\mathbf{v}} = \mathbf{U}^T\mathbf{v} \tag{7}$$

$$\text{IGFT} : \mathbf{v} = \mathbf{U}\bar{\mathbf{v}}. \tag{8}$$

2) FILTERING FACE NORMALS

Several works suggest filtering the normals instead of the vertices since they are considered as reliable indicators of geometric features (e.g., corner or edges). For example, an edge is indicated by a large difference between the normals of its two incident faces. The bilateral filter output of

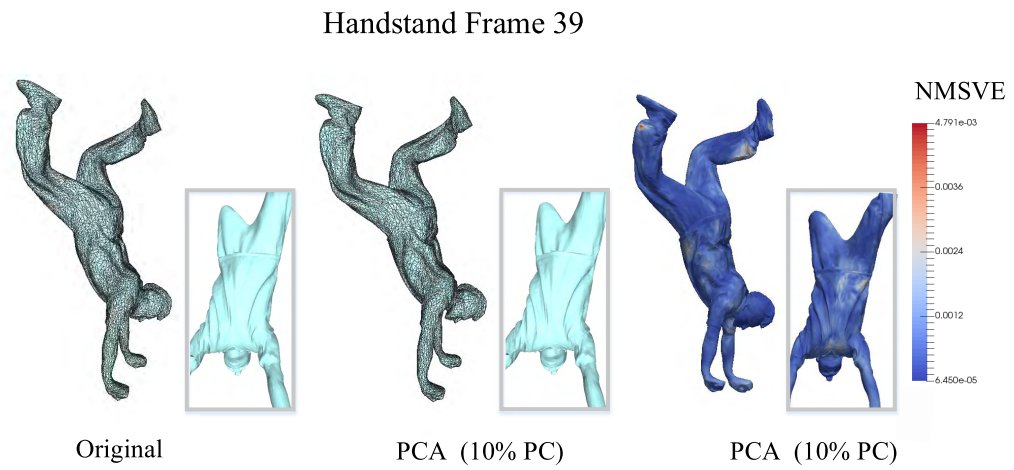


FIGURE 5. Left: The original mesh of the 39-th frame of the Handstand animation model. Middle: The PCA-based reconstructed mesh using 10% of the principal components. Right: Heatmap visualization of NMSVE to the handstand surface with maximum error equal to $4.79e^{-3}$.

the n_f face normals $\mathbf{n}_i$, $i = 1, \ldots, n_f$, using a normal guidance unit vector $\mathbf{g}_i$ that is associated with face f_i and is calculated as a weighted average of normals in a neighborhood of i, is computed by:

$$\hat{\mathbf{n}}_i = \frac{1}{W_i} \sum_{f_j \in \mathcal{N}_{f_i}} A_j K_s \left(\mathbf{c}_i, \mathbf{c}_j\right) K_r \left(\mathbf{g}_i, \mathbf{g}_j\right) \mathbf{n}_j \tag{9}$$

where $\mathcal{N}_{f_i} = \left\{f_j \,|f_j \text{ is adjacent to } f_i|\right\}$ is the set of faces in a neighborhood of f_i, A_j is the area of face f_j, W_i is a weight that ensures that $\hat{\mathbf{n}}_i$ is a unit vector and K_s, K_r are the spatial and range Gaussian kernels defined in Eqs. (10) and (11). More specifically, K_s and K_r decrease monotonically with respect to the distance of the centroids $\mathbf{c}_i$ and $\mathbf{c}_j$, and with the proximity of the guidance normals $\mathbf{g}_i$, $\mathbf{g}_j$ that lie on the unit sphere:

$$K_s \left(\mathbf{c}_i, \mathbf{c}_j\right) = e^{\left(-\frac{\|\mathbf{c}_i - \mathbf{c}_j\|_2^2}{2\sigma_s^2}\right)} \tag{10}$$

$$K_r \left(\mathbf{g}_i, \mathbf{g}_j\right) = e^{\left(-\frac{\|\mathbf{g}_i - \mathbf{g}_j\|_2^2}{2\sigma_r^2}\right)}, \tag{11}$$

where σ_s, σ_r are variance parameters. The output normals are then used to update the vertices in order to match the new normal directions $\mathbf{n}_i$, according to the iterative approach proposed in [19]. More specifically, the vertex positions $\hat{\mathbf{v}}_{i_1}$, $\hat{\mathbf{v}}_{i_2}$, $\hat{\mathbf{v}}_{i_3}$ of a face f_i are updated by:

$$\hat{\mathbf{v}}_{i_j}^{(t+1)} = \hat{\mathbf{v}}_{i_j}^{(t)} + \frac{1}{|\mathcal{F}_{ij}|} \sum_{z \in \mathcal{F}_{ij}} \hat{\mathbf{n}}_z \left[\hat{\mathbf{n}}_z^T \left(\hat{\mathbf{c}}_i^{(t)} - \hat{\mathbf{v}}_{i_j}^{(t)}\right)\right] \tag{12}$$

$$\hat{\mathbf{c}}_i^{(t)} = \left(\hat{\mathbf{v}}_{i_1}^{(t)} + \hat{\mathbf{v}}_{i_2}^{(t)} + \hat{\mathbf{v}}_{i_3}^{(t)}\right)/3 \tag{13}$$

where (t) is the iteration number, $\mathcal{F}_{i_j}$ is the index set of incident faces for $\hat{\mathbf{v}}_{i_j}$. This iterative process corresponds to a gradient descent process, applied for identifying the minimum energy cost across all faces, defined by:

$$\sum_{z \in \mathcal{F}_{i_j}} \left|\hat{\mathbf{n}}_z^T \left(\hat{\mathbf{c}}_i^{(t)} - \hat{\mathbf{v}}_{i_j}^{(t)}\right)\right|^2, \; j = 1, 2, 3. \tag{14}$$

This term penalizes any displacement which is perpendicular to the tangent plane defined by the vertex position $\hat{\mathbf{v}}_{i_j}^{(t)}$ and the local surface normal $\hat{\mathbf{n}}_z$.

C. DYNAMIC GEOMETRIES: LOW-RANK APPROXIMATIONS AND PCA BASED PROCESSING

A dynamic mesh is defined as a series of k static, mainly triangular, meshes representing a 3D animation, as shown in Fig. 1 (b). These meshes $\mathbf{M}_1, \ldots, \mathbf{M}_k$, namely poses or frames, are represented as in the static case, by a set of vertices $\mathcal{V}_i$ and a set-indexed faces $\mathcal{F}_i$. The Euclidean coordinates of n vertices for each frame i are used to form the following matrix:

$$\mathbf{M}_i = [\mathbf{v}_{1i}, \ldots, \mathbf{v}_{ni}] \in \Re^{3 \times n} \quad \forall \, i = 1, \ldots, k \tag{15}$$

The animation matrix $\mathbf{M} = \left[\mathbf{M}_1^T, \ldots, \mathbf{M}_k^T\right]^T \in \Re^{3k \times n}$ is constructed from the k sequential frames. Common Dynamic mesh compression approaches are based on the application of Principal Components Analysis (PCA) to the animation matrix $\mathbf{M}$ [20]–[22]. Considering that $\mathbf{M} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ corresponds to the SVD of the animation matrix, it is reasonable to assume that $\mathbf{M}$ can be adequately approximated by using fewer principal components, due to the underlying spatiotemporal coherences. To be more specific, the animation matrix $\hat{\mathbf{M}}$ can be reconstructed from $m < \min(3k, n)$ principal components as shown below:

$$\mathbf{M} \approx \hat{\mathbf{M}} = \mathbf{U}\hat{\mathbf{\Lambda}}_m\mathbf{U}^T. \tag{16}$$

In Fig. 5 we present the reconstructed mesh corresponding to the 39-th frame of the Handstand animation model [23],

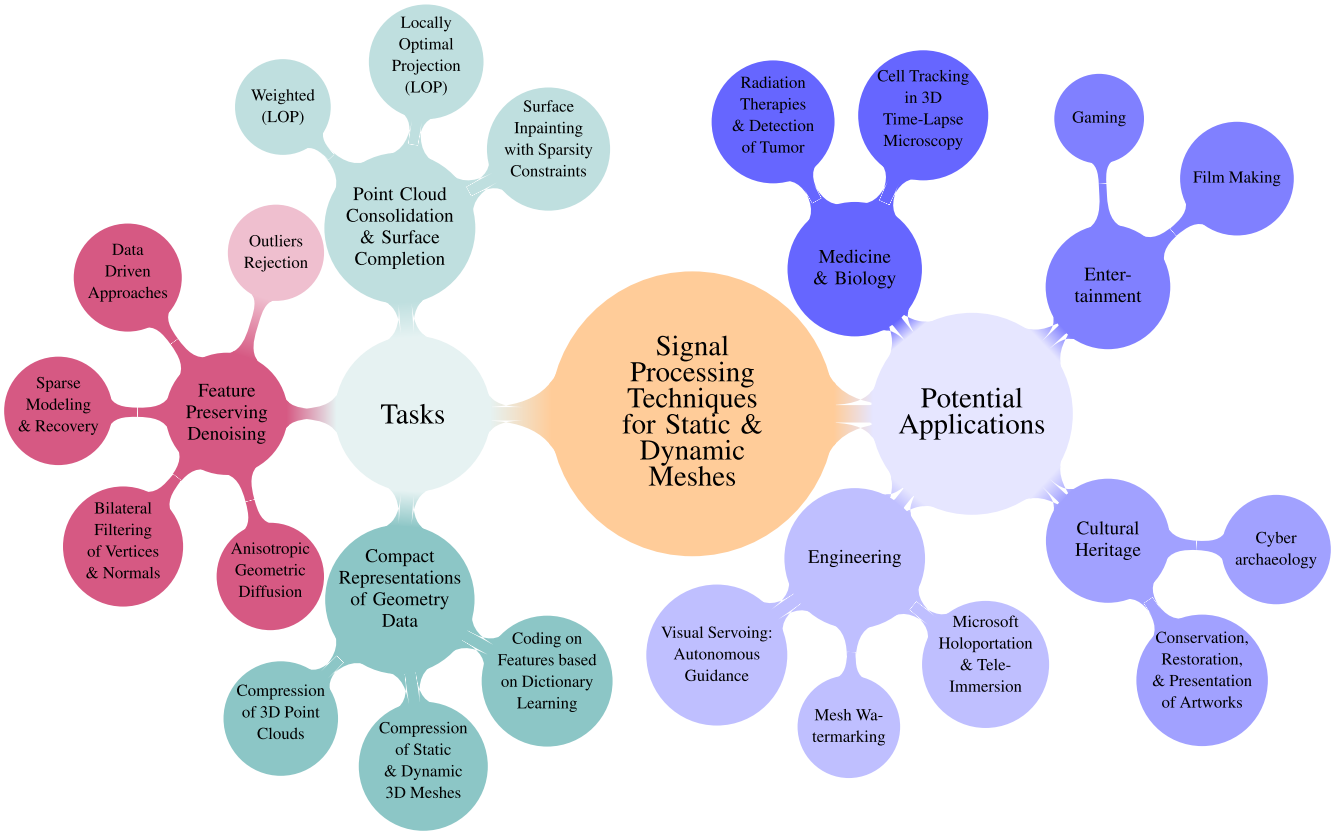


FIGURE 6. Static and Dynamic 3D Models: Signal processing tasks and applications. This figure illustrates some well-known approaches, separated in categories, which use both classical and modern SP techniques and they have been proposed for dealing with important geometry processing tasks and applications.

where only the 10% of the principal components have been used, i.e., $m = 50$ with $\mathbf{M}_{39} \in \Re^{525\times 10002}$. By inspecting the heatmap visualization of the mean square visual error [24], defined as the average error in the Cartesian and Laplacian domain, it is obvious that the reconstruction is not exactly equal to the input data, however, the loss of precision is not easily distinguishable.

In several dynamic meshes capturing soft-body animations, matrix $\mathbf{M} \in \Re^{3k\times n}$ can be accurately approximated by a low-rank matrix, introducing small reconstruction errors that cannot be easily perceived [25].

III. SPARSE MODELING FOR 3D MESHES IN A NUTSHELL

Sparsity is a fundamental characteristic in many sciences. Sparse modeling and optimization approaches have been widely adopted in many information sciences and are motivated by the classical problem of selecting a small number of predictive variables in high-dimensional datasets. In machine learning, the sparsity principle is used to automatically select a simple model among a large collection of them [26], while in signal processing, sparse modeling is employed to capture the ability of data to be expressed as linear combinations of a few atoms from a pre-described dictionary [27]. These models provide reasonable ways for exploiting the rich spatiotemporal structure, of the captured static and dynamic 3D geometry, using various statistical learning paradigms and well-documented merits, including PCA [20], [22], Dictionary Learning [28], Compressive Sampling (CS) [21], [29], [30] and Matrix Completion (MC) [31], to name a few.

Taubin [10] first suggested the application of Laplacian operators to the mesh vertex coordinates for discrete geometry processing. This analysis was motivated by the similarities between the spectral method with respect to mesh Laplacian and the classical Fourier approach. Several other studies, including implicit mesh fairing [32] and geometry compression [24], [33] are based also on the fact that the spectrum of smooth geometries is dominated by low-frequency components. During the last years, numerous approaches using both classical and modern SP techniques have been proposed for dealing with important geometry processing tasks (see Fig. 6), effectively exploiting correlations and sparsity structures that are hidden into the complex geometric patterns. The most important of these tasks are briefly presented in the following part of this section.

POINT CLOUD CONSOLIDATION

The output of a 3D scanning process is a sequence of unstructured point clouds, that are do not always sample uniformly

the surface and they usually are contaminated with noise. The process of addressing the aforementioned artifacts and enhancing these low-quality point cloud data is also known as *consolidation* [34]. Consolidation techniques [35]–[37] process the captured data and they generate a new point set which represents more accurately the *underlying shape*. They are without a doubt, an essential pre-processing step before surface reconstruction, where a well-sampled point cloud is transformed into a surface mesh. Despite the recent advances in the area of surface reconstruction [3], [38], [39], it is still a process that is highly susceptible to artifacts as those mentioned above. Specifically, decoupling the processes of point cloud consolidation and surface reconstruction can effectively avoid premature and erroneous decisions. In [40] a new algorithm is proposed for completing missing surface patches building on the same line of thought with sparse modeling and optimization techniques. The key intuition is that the spatiotemporal geometric coherence's that are present in any mesh, can be expressed as a sparse representation in some transformed domain. Arvanitis *et al.* [41] present a method for performing real-time outlier detection and removal, in time-varying point clouds using RPCA. Despite the promising results, no attention has been given to the exploitation of temporal coherence, thus limiting significantly its performance as compared to the methods that exploit this property. Regarding completion of missing parts, Zhong and Qin [40] use sparsity constraints for in-painting 3D surfaces with missing parts. The main drawback of the proposed approach is the fact that requires an eigenbasis decomposition which is very time consuming, especially in dense meshes. Laplacian interpolation is a well-known technique that has been successfully applied in image processing providing remarkably good results even in cases where a large amount of data is missing. Despite its computational efficiency when applied to point clouds, the reconstruction results are usually smoothed. To overcome this limitation, Arvanitis *et al.* [38] introduced a novel approach using the motion vectors instead of points. Although the method offers improved accuracy, its application in real scenarios has certain limitations since prior knowledge of a single pose is required.

FEATURE PRESERVING SURFACE DENOISING

Surface denoising approaches focus on eliminating the noise, preserving at the same time surface areas with geometric features of different scale. Despite the computational efficiency of the smoothing approaches, the lack of selectivity between salient features and noise is limiting, e.g., a noisy cube could become extremely rounded. In smoothing, the aim is to remove high-frequency information in the surface, whereas in denoising, the aim is to preserve genuine information at all frequencies by locally adjusting vertex positions while respecting the underlying features. The denoising approaches and can be classified into four major categories: (i) anisotropic geometric diffusion [42], [43] (ii) Bilateral Filtering of Vertices and Normals [12], [44], [19] (iii) Sparse modeling and recovery of geometric features in the Laplacian domain [45], [46] (iv) Data-driven approaches [47] that efficiently exploit training sets of noisy objects. Although the aforementioned methods successfully preserve sharp geometric features, they fail to preserve medium and/or small-scale features. More importantly, while their performance is significantly deteriorated when the surface is contaminated with complex noise patterns (e.g., anisotropic noise, real-scan noise). Finally, they cannot be used for reconstructing noisy 3D models, where poorly-shaped triangles are formed by vertices mostly lying on sharp edges. Thus, the main need for robust and fast algorithms able to manage dense dynamic meshes affected by complex noise seems to remain a challenge, offering ample opportunities for signal processing research.

COMPRESSION OF STATIC AND DYNAMIC GEOMETRIES

Geometry compression is much more demanding than connectivity compression, since the corresponding memory requirements are on average five times higher. This is attributed to the floating point representation for the raw geometry data. Thus, although state-of-the-art connectivity encoders are extremely effective, the compression of geometry information bring tough challenges to be solved. Lalos *et al.* [21] utilize sparse coding to generate compact representations of static geometries. Similarly, Yoon *et al.* [28] apply dictionary learning to get a few representative features that can be used to identify local patches from large-scale geometry data. Thanou *et al.* [48] and Lalos *et al.* [22] address the problem of compression of 3D point cloud sequences that are characterized by moving 3D positions. The fundamental theories presented in this tutorial will be essential for achieving improved compression rates offering at the same time low computational complexity [49].

While the major role of computer science disciplines (such as graph theory and data structures) in digital geometry processing is undeniable, the nature and scope of this field is certainly multidisciplinary and welcomes SP expertise and its recent advances, for effectively exploiting correlations and sparsity structures of different static and moving geometric data. This is also obvious when inspecting Table 1, where we present several low level information processing approaches that study the impact of sparse modeling and optimization tools to several static and dynamic 3D mesh processing tasks. The tremendous computing resources required due to the high spatiotemporal resolution of processing massive geometric datasets, bring tough challenges to be solved that become even more demanding in real time scenarios (the data are processed when they are received), such as large scene 3D acquisition and streaming (e.g., aerial 3D scanning, immersive communications), where decentralized processing is essential.

IV. SPARSE MODELING FOR GEOMETRY PROCESSING

The wide availability of inexpensive depth cameras (e.g., the Microsoft Kinect, Asus Xtion or PMD CamBoard)

TABLE 1. Prior art on exploiting the rich spatiotemporal structure, of the captured static and dynamic 3D geometry, using sparse modeling and optimization tools.

		Tools			Characteristics				
App.	Methods	Sparsity Exploitation	Low-rank Approximation	Sparsity Exploitation + Nuclear Norm Minimization	Static	Dynamic	Centralized	Decentralized	Real time
Outliers Detection	Wolff et al. [50]	✗	✗	✗	✓	✗	✗	✗	✗
	Arvanitis et al. [41]	✓	✓	✓	✓	✗	✓	✗	✓
	Brown [51]	✗	✗	✗	✓	✗	✗	✗	✗
	Mattei and Castrodad [52]	✓	✓	✓	✓	✗	✓	✗	✗
	Campbell [53]	✗	✓	✗	✓	✗	✗	✗	✗
	Candès et al. [54]	✗	✓	✗	✓	✗	✓	✗	✗
	Bayram et al. [55]	✗	✗	✗	✓	✗	✓	✗	✗
	Ngo et al. [56]	✗	✗	✗	✓	✗	✓	✗	✗
	Yang et al. [57]	✓	✗	✗	✓	✗	✓	✓	✗
	Bustos et al. [58]	✗	✗	✗	✓	✗	✓	✗	✗
	Vieira et al. [59]	✗	✗	✗	✓	✗	✓	✗	✗
Point Cloud Consolidation	Lipman et al. [35]	✗	✗	✗	✓	✗	✓	✓	✗
	Liu et al. [36]	✗	✗	✗	✓	✓	✓	✓	✗
	Berger et al. [3]	✓	✗	✗	✓	✗	✓	✗	✗
	Arvanitis et al. [38]	✓	✗	✓	✗	✓	✓	✗	✓
	Arvanitis et al. [39]	✗	✓	✗	✗	✓	✓	✗	✓
	Zhong and Qin [40]	✓	✗	✗	✓	✗	✓	✓	✗
	Xiong et al. [60]	✓	✗	✗	✓	✗	✓	✓	✓
	Mattei and Castrodad [52]	✓	✓	✓	✓	✗	✓	✗	✗
	Li et al. [61]	✗	✗	✗	✓	✗	✓	✗	✗
	Wang et al. [62]	✗	✗	✗	✓	✗	✓	✗	✗
Denoising	Hildebrandt and Polthier [42]	✗	✗	✗	✓	✗	✓	✗	✗
	Bajaj and Xu [43]	✗	✗	✗	✓	✗	✓	✗	✗
	Fleishman et al. [12]	✗	✗	✗	✓	✓	✓	✓	✓
	Zhang et al. [44]	✗	✗	✗	✓	✓	✓	✓	✓
	Sun et al. [19]	✗	✗	✗	✓	✓	✓	✓	✓
	Wang et al. [45]	✓	✗	✗	✓	✗	✓	✗	✗
	He and Schaefer [46]	✓	✗	✗	✓	✗	✓	✗	✗
	Wang et al. [47]	✗	✗	✗	✓	✗	✓	✓	✓
	Moench et al. [63]	✗	✗	✗	✓	✗	✓	✗	✓
	Arvanitis et al. [16]	✗	✗	✗	✓	✓	✓	✓	✓
	Bingqian et al. [64]	✓	✗	✗	✓	✗	✓	✓	✓
	Sarkar et al. [65]	✓	✓	✓	✓	✗	✓	✗	✗
Compression	Lalos et al. [21]	✓	✗	✗	✓	✗	✓	✗	✓
	Yoon et al. [28]	✓	✗	✗	✓	✗	✓	✗	✗
	Thanou et al. [48]	✓	✗	✗	✗	✓	✓	✗	✗
	Lalos et al. [49]	✓	✓	✗	✗	✓	✓	✗	✓
	Sorkine [33]	✓	✗	✗	✓	✗	✓	✗	✗
	Karni and Gotsman [24]	✓	✓	✗	✗	✓	✓	✗	✗
	Mamou et al. [66]	✓	✗	✗	✓	✗	✓	✗	✓
	Castorena et al. [67]	✓	✗	✗	✓	✗	✓	✗	✗
	Hou et al. [68]	✓	✓	✓	✓	✓	✓	✗	✓
	Hou et al. [69]	✓	✓	✓	✓	✓	✓	✗	✓

have facilitated the real-time generation of static and dynamic 3D Meshes, opening up a variety of mixed reality applications, such us bi-directional telepresence tools that offer perceptually enriched experiences, autonomous guidance for robots that reconstruct and respond rapidly to their environment [70] and others, bringing a range of benefits to business and society in various ways. However, processing this massive output dataset in real time using a central processor and storage is often impossible. In addition, despite the continuous advances in new generation image sensors and scanning technologies, the acquired data are often corrupted by severe noise [45], [46], outliers [41], [50], high variations in point density [36], misalignment, and missing data (see Fig. 7).

This section contributes to the ongoing efforts in fast and reliable generation of dense 3D models representing real-world moving objects capturing a wide range of SP-relevant tasks [71], such as nuclear norm minimization, CS, and MC that effectively exploit spatiotemporal correlations and sparsity structures that are hidden into the complex geometric patterns. It offers scalable architectures and optimization algorithms while revealing fundamental insights into the various processing and implementation tradeoffs involved in several static and dynamic 3D processing tasks. More specifically, subsection IV-A focuses on the identification of outliers in static and dynamic 3D geometric data, subsection IV-B deals with surface holes & non uniformly sampled surfaces and subsection IV-C offers details related

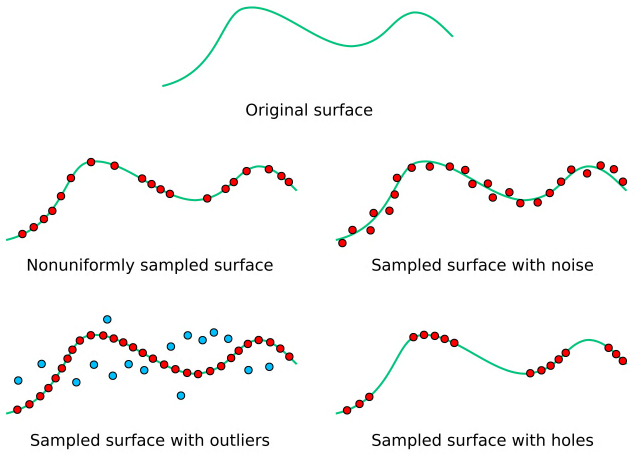


FIGURE 7. 3D acquisition artifacts shown in 2D, which are addressed using sparse modeling and optimization approaches.

to feature-aware denoising and compression of static and dynamic 3D models.

A. OUTLIER DETECTION IN STATIC AND DYNAMIC 3D SURFACE DATA

Commercially available 3D scanners generate a sequence of unstructured point clouds contaminated by a variety of acquisition errors including outliers, holes, surface noise, as those depicted in Fig. 7. In the rest subsection, we provide different consolidation approaches that exploit the sparsity of the outliers in the spatial domain.

l_1 MEDIAN PROJECTION

A common approach for dealing with outliers and noise is the l_1 median projection [51], a statistical tool that provides a robust global center of an arbitrary set of points. Assuming a set of data points $\mathbf{v} = \left[\mathbf{v}_{i_1}, \ldots, \mathbf{v}_{i_n}\right] \in \Re^{3\times n}$, the l_1 median $\mathbf{q}$ is obtained as the minimizer of the following cost function:

$$\mathbf{q} = \min_{\mathbf{x}} \left\{ \sum_{j=1}^{n} \left\| \mathbf{v}_{i_j} - \mathbf{x} \right\|_1 \right\} \tag{17}$$

where the l_1-norm is also known as the Least Absolute Deviations (LAD) and enforces sparsity in the solution.

Lipman *et al.* [35] apply a slide median filter locally, for evaluating a set of n' points $\mathbf{Q} = \left[\mathbf{q}_{i_1}, \ldots, \mathbf{q}_{i_{n'}}\right]$. More specifically, they suggest approximating the geometry of $\mathbf{P}$ by iteratively minimizing:

$$\mathbf{Q}^{(t+1)} = \min_{\mathbf{X}} \left\{ E_1\left(\mathbf{Q}^{(t)}, \mathbf{X}, \mathbf{v}\right) + E_2\left(\mathbf{Q}^{(t)}, \mathbf{X}\right) \right\} \tag{18}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n'}] \in \Re^{3\times n'}$, and:

$$E_1\left(\mathbf{Q}^{(t)}, \mathbf{X}, \mathbf{v}\right) = \sum_{i=1}^{n'} \sum_{j=1}^{n} \left\|\mathbf{x}_i - \mathbf{v}_j\right\|_2 \theta\left(\left\|\mathbf{q}_i^{(t)} - \mathbf{v}_j\right\|_2\right) \tag{19}$$

$$E_2\left(\mathbf{Q}^{(t)}, \mathbf{X}\right) = \sum_{z\in\mathcal{I}} \lambda_z \sum_{i\in\mathcal{I}\{z\}} \eta\left(\left\|\mathbf{x}_i - \mathbf{q}_z^{(t)}\right\|_2\right) \times \theta\left(\left\|\mathbf{x}_i - \mathbf{q}_z^{(t)}\right\|_2\right) \tag{20}$$

$\theta(r) = e^{-r^2/(h/4)^2}$, where h is a scalar value representing the size of the influence radius, $\mathcal{I} = [i_1, \ldots, i_{n'}]$ is the indices set, the term E_1 enforces the projected points to approximate $\mathbf{v}$, the term E_2 regularizes the points in $\mathbf{Q}^{(t)}$ by using local repulsion forces,[1] to ensure a fair spatial distribution and the weights $\lambda_z \; \forall \; z \in \mathcal{I}$ are penalty parameters that can be tuned, to trade off the robustness to outliers provided by $E_1(\cdot)$ with the fair distribution of points in space provided by $E_2(\cdot)$. The aforementioned approach is also known as Locally Optimal Projection algorithm (LOP). Huang *et al.* [29] have proposed the incorporation of locally adaptive weights into LOP, resulting in Weighted LOP (WLOP) resulting in an outlier free and uniformly distributed set of points.

OUTLIER DETECTION USING ROBUST PCA

An alternative approach would be to fit geometry data matrices with local point neighbours as a super position of a low-rank matrix capturing spatial coherence's and a sparse matrix that consist of outliers [52]. Building on the same line of thought, let $\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n] \in \Re^{3\times n}$ be the sequence of n noisy 3D points and $\mathbf{R}_i \in \Re^{n\times\kappa}$ represents an operator that extracts the κ-neighborhood of $\mathbf{v}_i$. Therefore, if $\mathbf{D}_i = [\mathbf{v}_i, \mathbf{v}_i\mathbf{R}_i] \in \Re^{3\times(\kappa+1)}$ is the local data matrix where its first column correspond to the 3×1 vector $\mathbf{v}_i$ and the next columns to the coordinates of the neighbors of $\mathbf{v}_i$, then the data matrix $\mathbf{D} = \left[\mathbf{D}_1^T, \mathbf{D}_2^T, \ldots, \mathbf{D}_n^T\right]^T \in \Re^{3n\times(\kappa+1)}$ can be decomposed as $\mathbf{D} = \mathbf{P} + \mathbf{S}$, where $\mathbf{P} \in \Re^{3n\times(\kappa+1)}$ is a low-rank matrix representing the space of the original points and their neighbors, while $\mathbf{S} \in \Re^{3n\times(\kappa+1)}$ is a sparse matrix representing the space where outliers lie. The conventional PCA estimates the $\mathbf{P}$ by seeking the best (in the least square sense) rank-m estimate of $\mathbf{P}$ by solving:

$$\min_{\mathbf{P}} \|\mathbf{D} - \mathbf{P}\|_F^2 \text{ subject to } \text{rank}(\mathbf{P}) \leq m. \tag{21}$$

Assuming that the few non zero entries of $\mathbf{S}$ have an independent and identically Gaussian distribution then the problem in Eq. (21) can be efficiently solved using the SVD. However, it is well known that standard PCA, is notoriously fragile to outliers, since its performance can significantly degrade in the presence of few corrupted samples, due to the quadratic error criterion used.

Robustification of PCA was initially achieved via robust estimates of the data covariance matrix [53]. Several relaxed optimization approaches with guaranteed convergence, have been lately proposed for recovering low-rank matrices in the presence of sparse and large errors [54], [72], that can be also applied for solving Eq. (21). Their goal is to estimate both data $\mathbf{P}$ and noise $\mathbf{S}$ by solving:

$$\min_{\mathbf{P},\mathbf{S}} \|\mathbf{P}\|_* + \lambda\|\mathbf{S}\|_1 \text{ subject to } \mathbf{P} + \mathbf{S} = \mathbf{D} \tag{22}$$

where λ a penalty parameter that can be tuned, to tradeoff the rank of matrix $\mathbf{P}$ and the degree of sparsity of outliers in $\mathbf{S}$. A widely adopted solver to the aforementioned convex problem is the Augmented Lagrange Multiplier (ALM) algorithm,

[1] A common repulsion function is given by $\eta(r) = 1/(3r^3)$

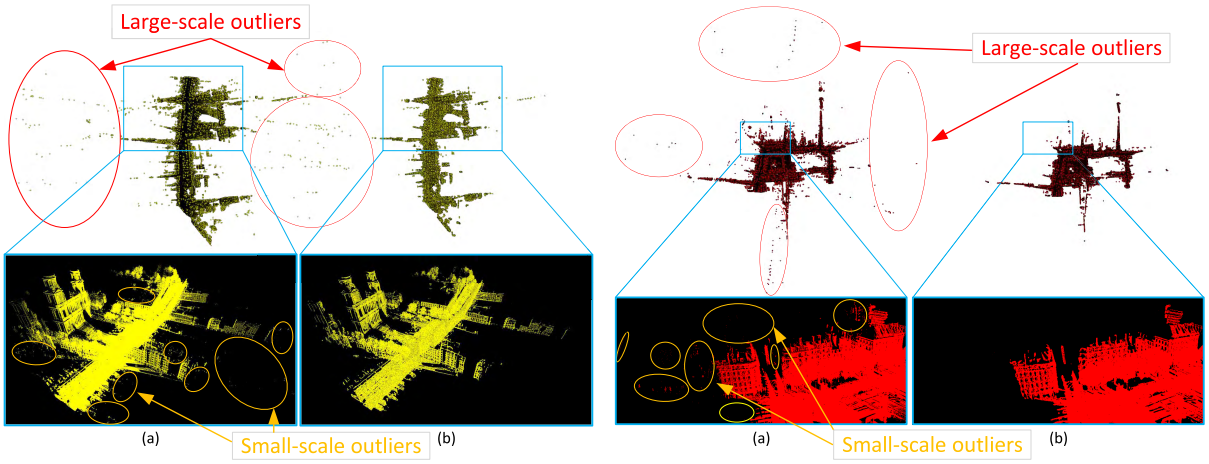


FIGURE 8. (a) Real-scanned urban environment with a great amount of outliers, (b) Point cloud without outliers after using the proposed approach.

which works with stability across a wide range of situations without the necessity of parameters configuration. The ALM method solves the problem of Eq. (22) by setting $(\hat{\mathbf{P}}, \hat{\mathbf{S}}) = \min_{\mathbf{P},\mathbf{S}} \mathcal{L}(\mathbf{P}, \mathbf{S}, \mathbf{Y})$ repeatedly, and then updating the Lagrange multiplier matrix via $\mathbf{Y}^{(t+1)} = \mathbf{Y}^{(t)} + \mu(\mathbf{D} - \mathbf{P}^{(t)} - \mathbf{S}^{(t)})$. In case studies, where a low-rank and sparse decomposition problem exists, both the $\min_{\mathbf{P}} \mathcal{L}(\mathbf{P}, \mathbf{S}, \mathbf{Y})$ and $\min_{\mathbf{S}} \mathcal{L}(\mathbf{P}, \mathbf{S}, \mathbf{Y})$ have very simple and efficient solutions. We introduce two ancillary operators $\mathcal{Q}_\tau$ and $\mathcal{D}_\tau$. $\mathcal{Q}_\tau : \Re \rightarrow \Re$ denotes the shrinkage operator $\mathcal{Q}_\tau(\cdot) = \text{sgn}(\cdot)\max(|\cdot| - \tau, 0)$ and extends it to matrices by applying it to each element while the $\mathcal{D}_\tau(\cdot)$ denotes the singular value thresholding operator given by $\mathcal{D}_\tau(\cdot) = \mathbf{U}\mathcal{Q}_\tau(\mathbf{\Sigma})\mathbf{V}^T$. The estimation of the point coordinates and the outliers is given by iteratively minimizing $\mathcal{L}$ with respect to $\mathbf{P}$ (fixing $\mathbf{S}$) and then with respect to $\mathbf{S}$ (fixing $\mathbf{P}$):

$$\min_{\mathbf{S}} \mathcal{L}(\mathbf{P}, \mathbf{S}, \mathbf{Y}) = \mathcal{Q}_{\lambda\mu^{-1}}(\mathbf{D} - \mathbf{P} + \mu^{-1}\mathbf{Y}) \tag{23}$$

$$\min_{\mathbf{P}} \mathcal{L}(\mathbf{P}, \mathbf{S}, \mathbf{Y}) = \mathcal{D}_{\mu^{-1}}(\mathbf{D} - \mathbf{S} + \mu^{-1}\mathbf{Y}) \tag{24}$$

Finally, the Lagrange multiplier matrix $\mathbf{Y}$ is updated using the residual $\mathbf{M}-\mathbf{P}-\mathbf{S}$. The process is repeated every F sequential frames.

$$\mathbf{S} = \mathbf{Y} = 0; \mu > 0 \tag{25}$$

$$\mathbf{P}^{(t+1)} = \mathcal{D}_{\mu^{-1}}(\mathbf{D} - \mathbf{S}^{(t)} + \mu^{-1}\mathbf{Y}^{(t)}) \tag{26}$$

$$\mathbf{S}^{(t+1)} = \mathcal{Q}_{\lambda\mu^{-1}}(\mathbf{D} - \mathbf{P}^{(t+1)} + \mu^{-1}\mathbf{Y}^{(t)}) \tag{27}$$

$$\mathbf{Y}^{(t+1)} = \mathbf{Y}^{(t)} + \mu(\mathbf{D} - \mathbf{P}^{(t+1)} + \mathbf{S}^{(t+1)}). \tag{28}$$

The singular value thresholding operation is the most computationally expensive operation, since it requires the singular value decomposition of $\mathbf{D} - \mathbf{S}^{(t)} + \mu^{-1}\mathbf{Y}^{(t)}$ that correspond to the singular values that are higher than μ^{-1}.The performance of the aforementioned approach, in terms of accuracy and complexity is determined by the stopping criterion (e.g., $\|\mathbf{D} - \mathbf{P} - \mathbf{S}\|_F \leq 10^{-7}\|\mathbf{D}\|_F$) and the number of the required principal eigenvectors determined by μ, which can be empirically estimated using the following rule $\mu = m^2/4\|\mathbf{D}\|_1$ [54].

At this point, it should be noted that the points in the component $\mathbf{P}_t$, where t is the number of executed iterations, correspond to the denoised version of $\mathbf{D}$. However the denoised PC consists of a collection of independent local estimates since each row i contains the estimation of $\mathbf{v}_i$ and its neighbors $\mathbf{R}_i\mathbf{v}_i$. Thus, each point is a member of multiple overlapping neighborhoods. In order to efficiently exploit all the independent estimates for a single point, Mattei and Castrodad [52] formulate a convex problem that exploits the redundancies present in the overlapping neighborhoods. More specifically, the denoised vertices $\hat{\mathbf{v}}$ of the PC can be obtained as the solution of the following optimization problem:

$$\min_{\mathbf{v}} \sum_{i=1}^{n} \left\| \left(\mathbf{v}\mathbf{R}_i - \mathbf{P}_{\mathbf{v}_i}\right) \mathbf{W}_i \right\|_F^2 + \lambda \left\| \mathbf{v} - \mathbf{P}_{(:,1)}^T \right\|_F^2 \tag{29}$$

where $\mathbf{P}_{\mathbf{v}_i} \in \Re^{3\times\kappa}$ is a submatrix that is formed from the $3(i-1)+1, \ldots, 3i$ rows and the $2, \ldots, \kappa+1$ columns of $\mathbf{L}_\kappa$, $\mathbf{W}_i = \text{diag}\{w_{i1}, \ldots, w_{i\kappa}\}$ is a diagonal matrix with the weights $w_{ij} = e^{-\theta_{ij}^2/\sigma_\theta^2}$ and θ_{ij}, $j = 1, \ldots, \kappa$ in its main diagonal, which represent the angle between the point normals $\mathbf{n}_i$ and $\mathbf{n}_j$. Moreover, σ_θ is a user defined parameter representing the size of the influence radius and λ is a regularization parameter that controls the trade-off between the denoising strength and data fidelity. The solution of Eq. (29) is given by:

$$\hat{\mathbf{v}} = \left(\lambda\mathbf{P}_{(:,1)}^T + \sum_{i=1}^{N} \mathbf{P}_{v_i}\mathbf{W}_i\mathbf{W}_i^T\mathbf{R}_i^T\right)\left(\lambda\mathbf{I} + \sum_{i=1}^{n} \mathbf{R_i}\mathbf{W}_i\mathbf{W}_i^T\mathbf{R}_i^T\right)^{-1} \tag{30}$$

where $\mathbf{I} \in \Re^{n\times n}$ is the unity matrix.

In Fig. 8, we show the results of our outliers removal process using, as input, two real-scanned and unorganized

point clouds containing an unknown percentage of outliers. Fig. 8 includes zoomed regions of these highly dense point clouds providing an easier evaluation of our method's performance. Two different types of outliers seem to be apparent: (i) the large-scale outliers (identified in red circles) which lie away from the point cloud and (ii) the small-scale outliers (identified in yellow circles) which are tangled with the useful information and mistakenly could be recognized as points. The proposed approach can efficiently remove both of these abnormalities, as shown in Fig. 8. The dataset, which is used for the evaluation of the outliers removal method, was taken from the ''IQmulus & TerraMobilita Contest'' benchmark [73]. It contains 10 highly dense point clouds ($13 \cdot 10^6 - 17 \cdot 10^6$ points) in different views of an urban environment.

OUTLIER DETECTION IN TIME VARYING PCs: DYNAMIC ROBUST PCA

In the time varying scenario, we consider a sequence of PCs (e.g., time varying point clouds (TVPC)), where each one of them is represented by a frame. After finding correspondences in temporally successive PCs we are able to formulate a matrix $\mathbf{M}$, where each column i represents the PCs of a frame i with authentic and corrupted points that are far from the model's surface and which we characterize as outliers. In the time varying case, $\mathbf{M} = \left[\mathbf{M}_1^T, \ldots, \mathbf{M}_k^T\right]^T \in \Re^{3k \times n}$ is defined as a series of PCs now corresponding to the different frames k. Our focus, is the identification of a low dimensional subspace $\mathbf{P}$ that captures as much variance of the authentic points in space and time using the Robust PCA (RPCA) approach presented above. We start by assuming that the TVPC is represented as a matrix $\mathbf{M}$, that can be decomposed as:

$$\mathbf{M} = \mathbf{P} + \mathbf{S} \tag{31}$$

where $\mathbf{P}$ is a low-rank matrix representing the space of the authentic points while $\mathbf{S}$ is a sparse matrix representing the space where outliers in the different frames $i = 1, \ldots, k$ live. By executing iteratively the steps presented in Eqs. (25)-(28) we are able to estimate the k consecutive denoised point clouds $\hat{\mathbf{v}}(1), \ldots, \hat{\mathbf{v}}(k)$ that correspond to the k columns of the estimated component $\mathbf{P}$.

Fig. 9 shows the application to three reconstructed frames (1, 88 and 109) of the ''Handstand'' animation model. The heatmap visualizations of the NMSVE show that the presented outlier removal methods (e.g., (i) WLOP executed in a per frame basis, (ii) Static RPCA executed in a per frame basis and (iii) Dynamic RPCA) successfully identify and remove the outliers even in the extreme case where 20% of the points can be considered as outliers, randomly distributed over the range of 1/50 of the bounding box's diagonal length.

Fig. 10 shows the reconstructed models, detecting and removing outlier points, using different recent and relevant approaches (e.g., [36], [39]). The conventional RPCA approach seems to provide the best results, although it fails to remove outliers that appear in more dense areas.

FIGURE 9. Outlier removal on the Handstand animation model: (i) 20% noisy points, (ii) output of WLOP, (iii) output of Static RPCA, (iv) output of RPCA applied to the time-varying point cloud matrix (DRPCA). The proposed outlier removal method successfully identifies and removes the outliers even in the extreme case where 20% of the points can be considered as outliers, randomly distributed over the range of 1/50 of the bounding box's diagonal length.

In such cases, the approach presented in [39], outperform the others because it does not only remove the identified outliers but it also can be modified to remove some controversial points, without losing information.

B. COMPLETION OF TIME VARYING PCs

A time varying PC can be represented by a low-rank matrix, where its columns correspond to sequential point clouds (i.e., registered TVCP's). The low-rank property can be also exploited for mitigating non-uniformities in thickness and spacing, due to acquisition errors, occlusions, physical limitations of the scanners or misalignment of multiple scans. More specifically, the under-sampled or completely missed regions corresponding to surfaces that are invisible to the cameras (e.g., deep cavities and bifurcations), lead to reconstruction errors or too smooth surfaces.

MC [74] is a well established signal processing framework which has been extensively used with great success in several applications. In particular, MC has been successfully applied to several computer graphics and vision problems, such as

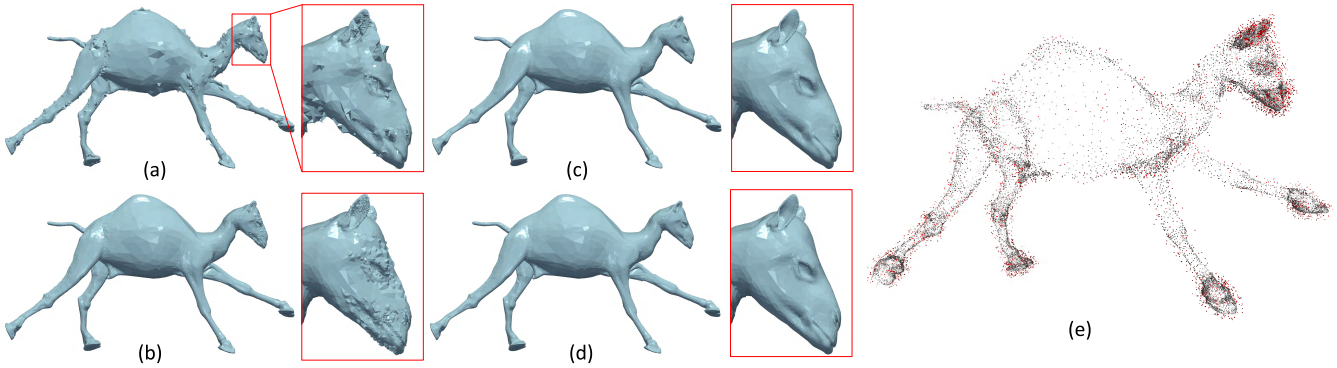


FIGURE 10. Reconstructed models using: (a) Liu-chan method in [36], (b) the low-rank component of RPCA, (c) the approach presented in [39], (d) the Original mesh, (e) the original model with 20% of outliers (Camel frame 36).

the recovery of occluded faces [75], the face image alignment [76], 3D mesh in painting [77], and the fusion of PCS from multiview images of the same object [78]. MC aims to the reconstruction of a low-rank matrix $\mathbf{M} \in \Re^{3k\times n}$ given a reduced number of known entries. In order to represent the known entries in matrix form, we define the sampling operator $\mathcal{P}_\Omega(\cdot)$ and the set of indices $\Omega \subseteq \{1,\ldots,3k\} \times \{1,\ldots,n\}$. Then, with $\mathcal{P}_\Omega(\mathbf{M})$ we denote the $3k \times n$ matrix that sets to zero the entries of $\mathbf{M}$ which are not in Ω, leaving unchanged the ones which are in Ω. A natural estimator for $\mathbf{M}$ would minimize the tradeoff between the least-squares error in fitting the known data and the rank [79], [80], e.g.,

$$\min_{\mathbf{X}} \text{rank}(\mathbf{X}) \text{ subject to } \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{M}). \tag{32}$$

Similarly to the RPCA approach, the nuclear norm is the closest convex approximation to the rank [74] and the original problem can be written as follows:

$$\min_{\mathbf{X}} \|\mathbf{X}\|_* \text{ subject to } \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{M}). \tag{33}$$

Note that, the minimization of the nuclear norm will result into the lowest rank reconstruction which satisfies the hard constraint of equality for the known values of $\mathbf{M}$.

► *Remark:* (on the projection operator and the Hadamard product): The projection operator $\mathcal{P}_\Omega(\cdot)$ can be also represented as a Hadamard product of the animation matrix and the matrix $\mathbf{E}_\Omega \circ \hat{\mathbf{M}}$ composed by ones and zeros, i.e.,

$$[\mathbf{E}_\Omega]_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \Omega \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

Recall that the Hadamard product can be expressed as:

$$\mathbf{E}_\Omega \circ \mathbf{M} = \sum_{i=1}^{3k} \mathbf{E}_{ii}\mathbf{M}\mathcal{D}(\mathbf{E}_{\Omega,i}) \tag{35}$$

where $\mathbf{E}_{ii}$ is the matrix with a unity value at the position (i, i) while $\mathcal{D}(\mathbf{E}^T_{\Omega,i})$ is the diagonal matrix with the i-th row of $\mathcal{P}_\Omega$. Based on the following property of the vectorization function:

$$\text{vec}(\sum_i \mathbf{A}_i\mathbf{B}\mathbf{C}_i) = \sum_i (\mathbf{C}_i^T \otimes \mathbf{A}_i)\text{vec}(\mathbf{B}) \tag{36}$$

we have that:

$$\text{vec}(\mathbf{E}_\Omega \circ \mathbf{M}) = \text{vec}(\sum_i \mathbf{E}_{ii}\mathbf{M}\mathcal{D}(\mathbf{E}_{\Omega,i})) \tag{37}$$

$$= \sum_{i=1}^{3k} (\mathcal{D}(\mathbf{E}_{\Omega,i})^T \otimes \mathbf{E}_{ii})\text{vec}(\mathbf{M}) \tag{38}$$

$$= \mathcal{D}(\mathcal{P}_\Omega)\text{vec}(\mathbf{M}) \tag{39}$$

This result indicates that the projection operator can be expressed as a simple linear matrix-vector product after vectorization.◄

Relaxing the hard constraint of Eq. (33), we can formulate the following unconstrained optimization problem:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{X} - \mathbf{M})\|_F^2 + \tau\|\mathbf{X}\|_*, \tag{40}$$

which jointly minimizes the Euclidean distance between the known points and the recovered, and imposes low-rank to the recovered geometry, depending on the weighting parameter τ. Although Eq. (40) is an unconstrained form, the nuclear norm term prohibits from a closed form solution. To overcome this issue, a standard approach is to split the problem into two parts and use alternating minimization techniques to iteratively solve each one. Specifically, we introduce an auxiliary matrix $\mathbf{Y} \in \Re^{3k\times n}$ and formulate the optimization problem of Eq. (40) as follows:

$$\min_{\mathbf{X},\mathbf{Y}} \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{M})\|_F^2 + \tau\|\mathbf{X}\|_* \text{ subject to } \mathbf{X} = \mathbf{Y}, \tag{41}$$

which can be efficiently solved by using the Alternating Direction Method of Multipliers (ADMM) [79].

► *Remark:* (on ADMM and relation with ALM): ADMM can be viewed as an approximate ALM, where the first step is replaced by two, splitting the original joint optimization over the primal variables. This splitting improves the decoupling of the problem and permits the distributed implementation of the technique. However, there is no general way of qualifying how close is this approximation to the joint one. More specifically, ADMM is composed by

three steps:

$$\mathbf{x}^{(t+1)} = arg\min_{\mathbf{x}} \mathcal{L}_a(\mathbf{x}, \mathbf{z}^{(t)}, \mathbf{y}^{(t)}) \quad (42)$$

$$\mathbf{z}^{(t+1)} = arg\min_{\mathbf{x}} \mathcal{L}_a(\mathbf{x}^{(t+1)}, \mathbf{z}, \mathbf{y}^{(t)}) \quad (43)$$

$$\mathbf{y}^{(t+1)} = \mathbf{y}^{(t)} + \mu(\mathbf{A}\mathbf{x}^{(t+1)} + \mathbf{B}\mathbf{z}^{(t+1)} - \mathbf{c}) \quad (44)$$

where again $\mu = m^2/4\|\mathbf{M}\|_1$ is initialized as in the static case. An important property that renders ADMM a distributed technique is that the order of the processes does not affect the steady-state performance. A pre-determined communication protocol is responsible for defining the way where the updates are being synchronized among them. ◄

Algorithm 1 ADMM-Based PC Completion (MC)

Require: $\Omega, \mathcal{P}_\Omega \circ \mathbf{M}, \rho$
Ensure: $\mathbf{X}_{I_{\max}}$
  for $t = 0, \ldots, I_{\max} - 1$ **do**
    $\mathbf{X}^{(t+1)} \leftarrow \mathcal{D}_{\tau/\rho}\left(\mathbf{Y}^{(t)} - \rho^{-1}\mathbf{Z}^{(t)}\right)$
    Solve the system to estimate $\mathbf{Y}^{(t+1)}$:
      $\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{M}) - \rho\left(\mathbf{X}^{(t+1)} - \mathbf{Y}\right) = \mathbf{Z}^{(t)}$
    $\mathbf{Z}^{(t+1)} \leftarrow \mathbf{Z}^{(t)} + \rho(\mathbf{X}^{(t+1)} - \mathbf{Y}^{(t+1)})$
  end for

GRAPH-BASED COMPLETION OF TVPC's

We have seen that the geometry data matrix has the low-rank property in the principal components domain. However, due to the special nature of the geometric data, correlations also exist in the 3D space due to the proximity of the 3D points, especially in flat areas. This information can be exploited and incorporated into the previous problem as an additional constraint, extending the developed tools to this case. In particular, if we consider a reduced number of 3D points forming an incomplete animation matrix, the proximity metric can be embedded into problem Eq. (40) according to the following formulation:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{X} - \mathbf{M})\|_F^2 + \tau\|\mathbf{X}\|_* + \frac{\gamma}{2}\|\mathbf{X}\mathbf{L}\|_F^2 \quad (45)$$

where the last term imposes that the positions of the neighboring nodes of the reconstructed frames will be close to each other. The parameter γ represent the associated regularization parameter of the graph Laplacian $\mathbf{L}$.

► *Remark:* In [81], a well known technique termed as Least-Square Meshes (LSM) which exploits the spatial coherence, has been successfully employed for the reconstruction of a static PC. In particular, LSM can be expressed as the solution of the following extended system of equations:

$$\mathbf{X}_{\text{LSM}}\left[\,\delta\mathbf{L}\ \mathbf{I}_{n\times K}\,\right] = \left[\,\mathbf{0}_{3k\times n}\ \mathbf{R}\,\right] \quad (46)$$

where $\mathbf{X}_{\text{LSM}} \in \Re^{3k\times n}$, $\mathbf{L} \in \Re^{n\times n}$, δ is a weighting parameter and $\mathbf{R} \in \Re^{3k\times K}$ are the known $3kK$ *anchor* points from $\mathbf{M}$ for $K \leq 3k$. Let $\mathbf{L}_e = \left[\,\delta\mathbf{L}\ \mathbf{I}_{n\times\kappa}\,\right]$ and $\mathbf{R}_e = \left[\,\mathbf{0}_{3k\times n}\ \mathbf{R}\,\right]$, then the least-squares solution of Eq. (46) is expressed as: $\mathbf{X}_{\text{LSM}} = \mathbf{R}_e\mathbf{L}_e^T(\mathbf{L}_e\mathbf{L}_e^T)^{-1}$. It is important to note that, the optimization problem of Eq. (45) can be expressed as Eq. (46) for $\tau = 0$, $|\mathcal{P}(\mathbf{M})| = 3kK$ and $\gamma = \delta$. ◄

The augmented Lagrangian of the equivalent splitting version of the optimization problem Eq. (45) is expressed as:

$$\mathcal{L}_a(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{M})\|_F^2 + \tau\|\mathbf{X}\|_* + \frac{\gamma}{2}\|\mathbf{Y}\mathbf{L}\|_F^2 + \frac{\rho}{2}\|\mathbf{X} - \mathbf{Y}\|_F^2 + \langle\mathbf{Z}, \mathbf{X} - \mathbf{Y}\rangle \quad (47)$$

where, as previously, $\mathbf{X}, \mathbf{Y}$ are the primal variables, $\mathbf{Z}$ the dual variable, and ρ is the penalizing factor. Taking the partial derivative with respect to the $\mathbf{Y}$ variable is expressed as:

$$\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{M}) + \gamma\mathbf{Y}(\mathbf{L}^T\mathbf{L}) + \rho\left(\mathbf{Y} - \mathbf{X}^{(t+1)}\right) - \mathbf{Z}^{(t)} = \mathbf{0}. \quad (48)$$

The minimizer of Eq. (48) is obtained by solving the following $3kn \times 3kn$ linear system:

$$\left[\sum_i \mathbf{B}_i^T \otimes \mathbf{A}_i\right]\text{vec}(\mathbf{Y}) = \text{vec}(\mathbf{C}) \quad (49)$$

where

$$\mathbf{A}_i = \begin{cases} \mathcal{P}_{ii} & i = 1, \ldots, 3k \\ \gamma\mathbf{L}^T\mathbf{L} & i = 3k+1 \\ \rho\mathbf{I}_n & i = 3k+2 \end{cases} \quad (50)$$

$$\mathbf{B}_i = \begin{cases} \text{diag}(\mathcal{P}_{\Omega,i}) & i = 1, \ldots, 3k \\ \mathbf{I}_n & i = 3k+1, 3k+2 \end{cases} \quad (51)$$

and $\mathbf{C} = \mathcal{P}_\Omega(\mathbf{M}) + \rho\mathbf{X}^{(t+1)} + \mathbf{Z}^{(t)}$, which has a unique solution for $\rho > 0$ [25].

Algorithm 2 Graph-Based Surface Completion (GMC)

Require: $\Omega, \mathcal{P}_\Omega \circ \mathbf{M}, \rho, \mathbf{L}$
Ensure: $\mathbf{X}_{I_{\max}}$
  for $t = 0, \ldots, I_{\max} - 1$ **do**
    $\mathbf{X}^{(t+1)} \leftarrow \mathcal{D}_{\tau/\rho}\left(\mathbf{Y}^{(t)} - \rho^{-1}\mathbf{Z}^{(t)}\right)$
    Solve the system (49) to estimate $\mathbf{Y}^{(t+1)}$:
      $\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{M}) + \gamma\mathbf{Y}(\mathbf{L}^T\mathbf{L}) + \rho\left(\mathbf{Y} - \mathbf{X}^{(t+1)}\right) = \mathbf{Z}^{(t)}$
    $\mathbf{Z}^{(t+1)} \leftarrow \mathbf{Z}^{(t+1)} + \rho(\mathbf{X}^{(t+1)} - \mathbf{Y}^{(t+1)})$
  end for

In Fig. 11 we depict three frames (1, 88 and 109) of the "Handstand" animation model. We compare the original, the LSM, the MS (Algorithm 1) and the GMC techniques (Algorithm 2) for the case of 30% undersampling ratio. A close look at the heatmap visualization of the normalized mean square visual error (NMSVE), we can easily observe that the exploitation of the underlying low-rank structure of the animation matrix (GMC) results in better reconstruction quality compared to MC (e.g., preservation of the geometric features of a different scale).

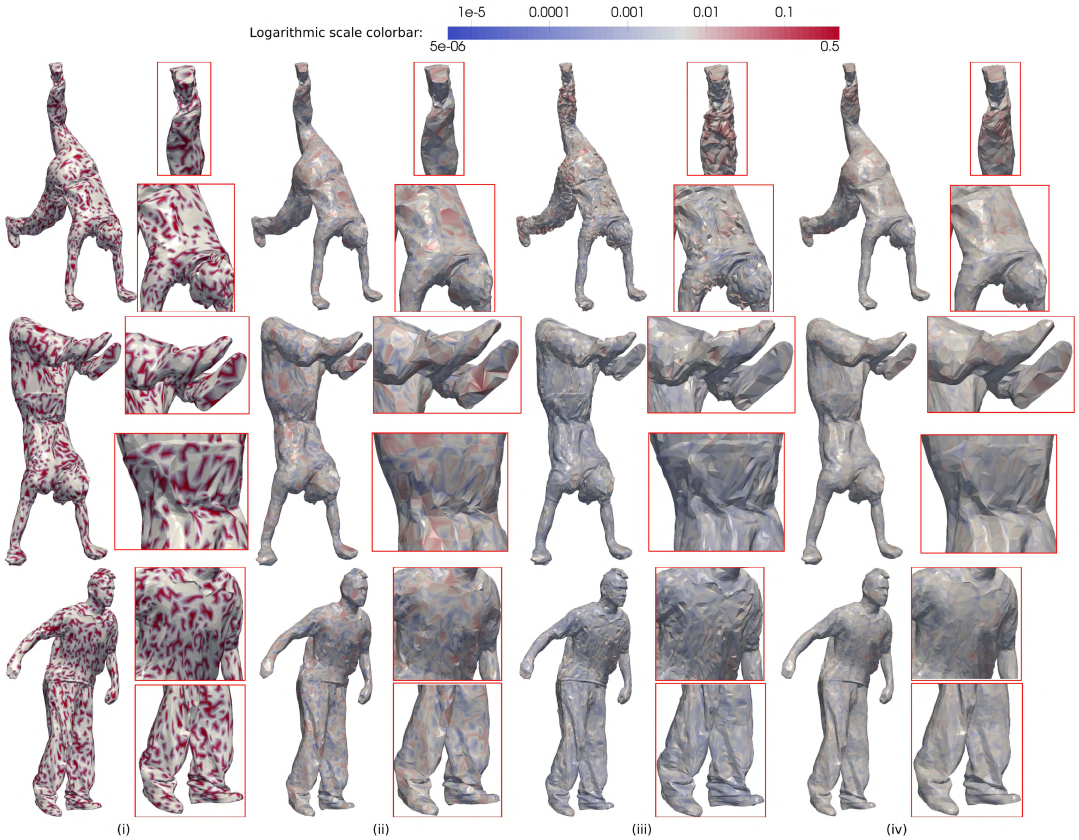


FIGURE 11. **Dynamic PC Completion Results: (i) Handstand model: known points are highlighted with red, (ii) reconstructed mesh using LSM approach, (iii) reconstructed mesh using MC approach, (iv) reconstructed mesh using GMC approach.**

C. FEATURE-AWARE MESH DENOISING AND COMPRESSION

Throughout the years, several approaches have been proposed for performing feature-preserving mesh denoising [19], [45]–[47]. In many cases small and large-scale geometric features cannot be easily differentiated from noise. Recent works focus on addressing this challenge, by exploiting the fact that geometric features can be considered as outliers in the Laplacian domain [83], [84].

GUIDED MESH NORMAL FILTERING

Zhang *et al.* [44] propose a new mesh normal filtering framework based on the bilateral filter presented in Section 2.B.2. More specifically, they first apply bilateral filtering to the face normals, using a properly constructed normal field as the guidance and then, the vertex positions are updated according to the filtered face normals via Eqs. (12), (13). The face normals of a 3D mesh (that is not corrupted by noise) can provide good guidance for joint bilateral filtering. However, in the presence of noise, the noisy normals become less reliable for indicating the features of the ground truth shape, eventually leading to erroneous results. For this reason, it seems necessary to compute guidance normal on a face, that provides a reliable estimation of the true normal even when the model is corrupted with severe noise.

For each face f_i of the triangle mesh, a lot of candidate patches are available. We define as a candidate patch $\mathcal{P}_i$ the union of f_i with all surrounding faces that share vertices with it. To compute the guidance normal at a face f_i, all patches that contain f_i are candidates, while the one with the most consistent normal direction is selected, based on a function which measures the consistency of the normals. The consistency of each patch $\mathcal{P}_i$ is measured by the function:

$$\begin{aligned}\mathcal{H}\left(\mathcal{P}_i\right) &= \Phi\left(\mathcal{P}_i\right)\cdot\mathcal{R}\left(\mathcal{P}_i\right)\\ \Phi\left(\mathcal{P}_i\right) &= \max_{f_k,f_j\in\mathcal{P}_i}\left\|\mathbf{n}_j-\mathbf{n}_k\right\|_2,\\ \mathcal{R}\left(\mathcal{P}_i\right) &= \frac{\max_{e_j\in E_{P_i}}\phi(e_j)}{\epsilon+\sum_{e_j\in E_{P_i}}\phi(e_j)},\end{aligned}$$

where E_{P_i} is the set of edges with both incident faces contained in patch P_i. The edge saliency $\phi(e_i)$ in a patch, is an important factor which determines the decision for the ideal patch's selection. This factor can be used for identifying the saliency of an edge e_j and can be estimated from the difference of the normal's corresponding to the two incident faces f_{i_1}, f_{i_2}:

$$\phi(e_i) = \|\mathbf{n}_{i_1} - \mathbf{n}_{i_2}\|_2 \tag{52}$$

The set of the normalized $\phi(e_i)$ from all of the candidate patches could be defined as:

$$\Psi(\mathcal{P}) = (\frac{\phi(e_1)}{\phi_{\max}}, \frac{\phi(e_2)}{\phi_{\max}}, \cdots, \frac{\phi(e_n)}{\phi_{\max}}) \tag{53}$$

where $\phi_{\max} = \max(\phi(e_i)) \; \forall \; i = 1, n$. Based on the fact that the l_1-norm for a vector is related to its sparsity, we can conclude that a small l_1-norm of $\Psi(\mathcal{P})$ indicates a small number of edges are much more salient than the others. Additionally, the l_1-norm of $\Psi(\mathcal{P})$, can be used for identifying the patches that can be used for the guidance normal estimation. Finally, the guided g_i normal is estimated as the area-weighted average normal of this patch and the bilateral procedure is applied on the guidance normals.

FEATURE IDENTIFICATION USING l_1 MINIMIZATION

The mesh vertices that are considered as features are sparse and their number is usually much smaller than non-feature vertices. Lu *et al.* [85] exploit the fact that the magnitude of the δ coordinates of feature vertices is much larger than the δ coordinates of the non-feature vertices, which usually tends to zero. Then, they defined a function J_i used for identifying whether the i-th vertex is a feature or not, based on the aforementioned observation. Building on the same line of thought, it is reasonable to define f_i as the dot product of the i-th vertex normal n_i and its Laplacian L_iP for the i-th vertex, as follows:

$$J_i = \mathbf{n}_i \cdot (\mathbf{L}_i \mathbf{v})^T \quad \forall \; i = 1, \ldots, n \tag{54}$$

where $\mathbf{L}_i$ is the i-th row of the matrix $\mathbf{L}$, and $\mathbf{v}$ is the vector composed with all the vertex positions. An extension of Eq. (54) providing more accurate results is the following:

$$J_i = \frac{1}{\sum_{j \in N_{f_i}} a_j} \sum_{j \in N_{f_i}} a_j (\mathbf{n}_i - \mathbf{n}_{\mathbf{c}j}) \cdot (\mathbf{L}_i \mathbf{v})^T \quad \forall \; i = 1, \ldots, n \tag{55}$$

where N_{f_i} is the first-ring neighboring faces of the i-th vertex, a_j is the j-th face area in N_{f_i}, and $\mathbf{n}_{\mathbf{c}j}$ is the corresponding face normal.

It is obvious that only few features exist in a model, and correspond to the vertices with large J_i values, while most vertices, correspond to flat areas and they take a small close-to-zero J_i value. Since, features are indeed sparse in a model, l_0-norm minimization can be used for identifying them [46]. To alleviate the computational complexity issue, a simple l_1-norm regularized formula could be used, given by:

$$\min \|\mathbf{x} - \mathbf{J}\|_2 + \lambda \|\mathbf{x}\|_1 \tag{56}$$

where $\mathbf{J} = [J_1, \cdots, J_n]^T$, $\mathbf{x}$ is the vector with all the unknowns, and λ is the regularization parameter to control the sparsity of the vector $\mathbf{x}$.

IDENTIFYING SHAPE FEATURES USING RPCA

To exploit the sparsity of features in the Laplacian domain we suggest working with guided normals defined as the average normal of an ideal neighborhood of faces for each face centroid. More specifically, we suggest following the idea originally proposed in [44] and construct a matrix with guided normals $\mathbf{A} \in \Re^{n_f \times 3\kappa}$, where n_f is the number of faces, as shown in Fig. 12. Then we suggest exploiting spatiotemporal coherences and the sparsity of the faces that are identified as features using the RPCA approach described by the steps presented in Eq. (25). The output of this approach is a set of guided normals that define piecewise smooth areas $\mathbf{P}_A$ and a set of normal corresponding to features $\mathbf{S}_A$. The results of an RPCA based denoising approach is shown in Fig. 14, where it is obvious that the sharp features are progressively identified as the number of RPCA iterations increases.

In Fig. 15, we show the results for different recent feature detection algorithms applied in original and noisy objects of different 3D models. The method in Fig. 15-(a) provides thicker areas of features, which is convenient in

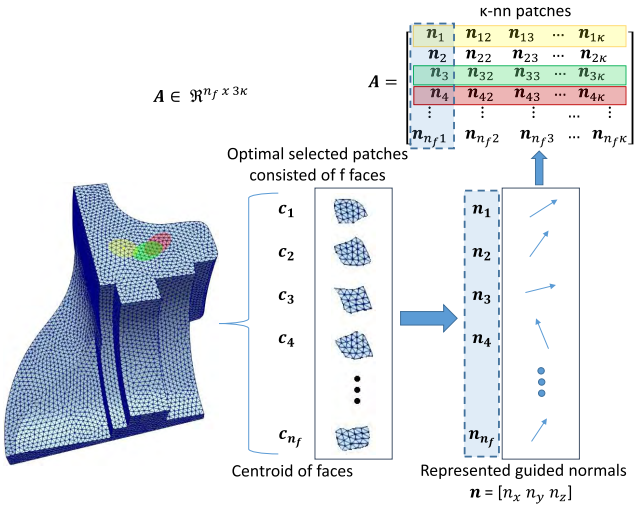


FIGURE 12. **Static case: Construction of a low-rank matrix.**

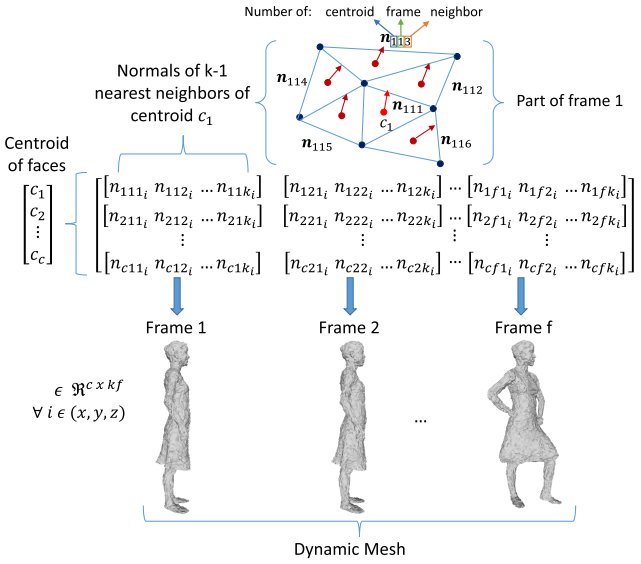


FIGURE 13. **Dynamic case: Construction of a low-rank matrix.**

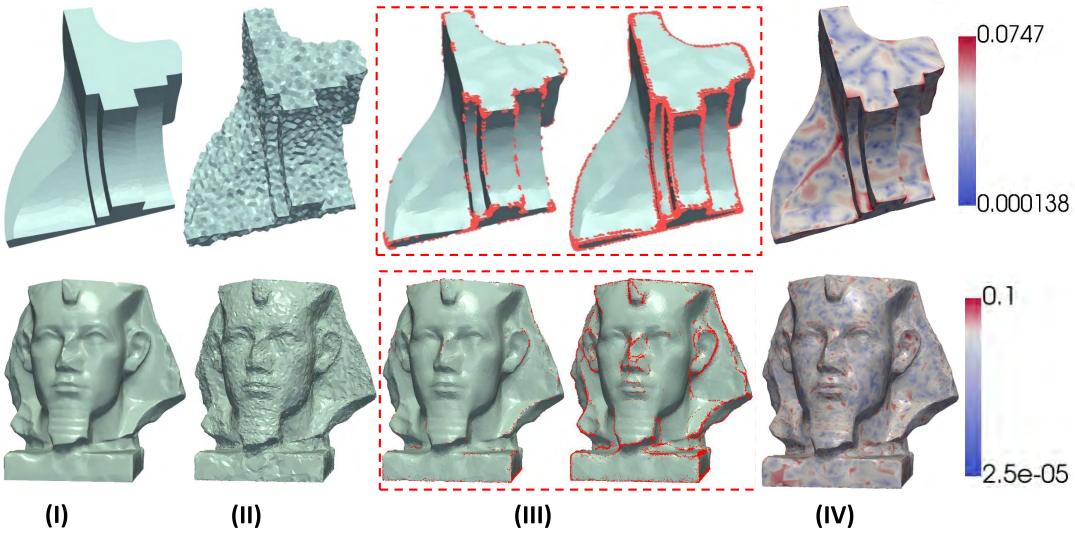


FIGURE 14. Mesh Denoising using RPCA on Fandisk and Farao model, (I) Original meshes, (II) Noisy meshes, (III) RPCA applied to the guided normals selected of the optimal overlapped patches, (IV) Heatmap visualization of difference between reconstructed and original mesh.

TABLE 2. Extensive evaluation of results using as metrics both the angle θ and the Hausdorff distance error (hde).

Name of Model	Zhang et al. [45]	Wang et al. [48]	Arvanitis et al. using Grouped Parameters [17]	Arvanitis et al. using Ideal Parameters [17]
Child	$\theta = 7.4941^o$ hde = 0.0039	$\theta = 6.9801^o$ hde = 0.0043	$\theta = 6.2145^o$ hde = **0.0038**	$\theta = \mathbf{6.1615^o}$ **hde = 0.0038**
Chinese lion	$\theta = 7.9365^o$ hde = 0.5559	$\theta = 7.6701^o$ hde = 0.5143	$\theta = 7.1285^o$ hde = 0.4916	$\theta = \mathbf{6.939^o}$ **hde = 0.4893**
Elephant	$\theta = 4.7512^o$ hde = 0.0036	$\theta = 3.751^o$ hde = 0.0028	$\theta = 3.4015^o$ **hde = 0.0025**	$\theta = \mathbf{3.288^o}$ **hde = 0.0025**
Octahedron	$\theta = 1.1468^o$ hde = 0.047	$\theta = 1.8062^o$ hde = 0.0459	$\theta = 0.8323^o$ hde = 0.0196	$\theta = \mathbf{0.6566^o}$ **hde = 0.0115**
Leg	$\theta = 3.1952^o$ hde = 0.1615	$\theta = 3.053^o$ hde = 0.1774	$\theta = 2.7617^o$ hde = 0.1547	$\theta = \mathbf{2.7588^o}$ **hde = 0.1523**
Julius	$\theta = 6.4266^o$ **hde = 0.0032**	$\theta = 6.5546^o$ hde = 0.0039	$\theta = 6.155^o$ hde = 0.0086	$\theta = \mathbf{6.0766^o}$ hde = 0.0086

TABLE 3. Extensive evaluation of results in comparison with a variety of different state-of-the-art methods using as metrics both the angle θ and the Hausdorff distance error (hde).

Name of Model	Fleisman et al. [13]	Jones et al. [14]	Sun et al. [20]	Zheng et al. (l) [15]	Zheng et al. (g) [15]	He and Schaefer [47]	Zhang et al. [45]	Arvanitis et al. [17]
block	$\theta = 12.7155^o$ hde = 0.8781	$\theta = 13.8501^o$ hde = 0.653	$\theta = 5.8023^o$ hde = 0.8206	$\theta = 5.3062^o$ hde = 0.7257	$\theta = 8.0165^o$ hde = 0.7009	$\theta = 4.9734^o$ hde = 0.6698	$\theta = 3.572^o$ hde = 0.71	$\theta = \mathbf{3.3026^o}$ **hde = 0.5869**
fandisk	$\theta = 22.4862^o$ hde = 0.2249	$\theta = 27.9264^o$ hde = 0.2093	$\theta = 13.1918^o$ hde = 0.1817	$\theta = 14.2553^o$ hde = 0.1864	$\theta = 15.0545^o$ hde = 0.1892	$\theta = 6.2186^o$ hde = 0.124	$\theta = 6.3721^o$ hde = 0.1388	$\theta = \mathbf{6.0669^o}$ **hde = 0.1208**
sphere	$\theta = 12.5796^o$ hde = 0.8106	$\theta = 17.3618^o$ hde = 0.477	$\theta = 11.892^o$ hde = 0.5514	$\theta = 6.7047^o$ **hde = 0.3398**	$\theta = 9.6974^o$ hde = 0.471	$\theta = 12.9566^o$ hde = 0.536	$\theta = 10.1697^o$ hde = 0.4607	$\theta = \mathbf{6.2407^o}$ hde = 0.3917
twelve	$\theta = 11.7204^o$ hde = 0.1357	$\theta = 11.093^o$ hde = 0.1074	$\theta = 7.4519^o$ hde = 0.0728	$\theta = 7.3683^o$ hde = 0.0858	$\theta = 7.271^o$ hde = 0.0717	$\theta = 8.4626^o$ hde = 0.1301	$\theta = 2.7542^o$ hde = 0.11	$\theta = \mathbf{2.6543^o}$ **hde = 0.0695**

some applications. A disadvantage of this method is that it requires a careful selection of parameters, making it practically difficult to use. The approaches in Fig. 15-(b),(c) provide perfect results when they are used in original/denoised meshes with intense features (e.g., Fandisk). However, the results are far from satisfying when they are

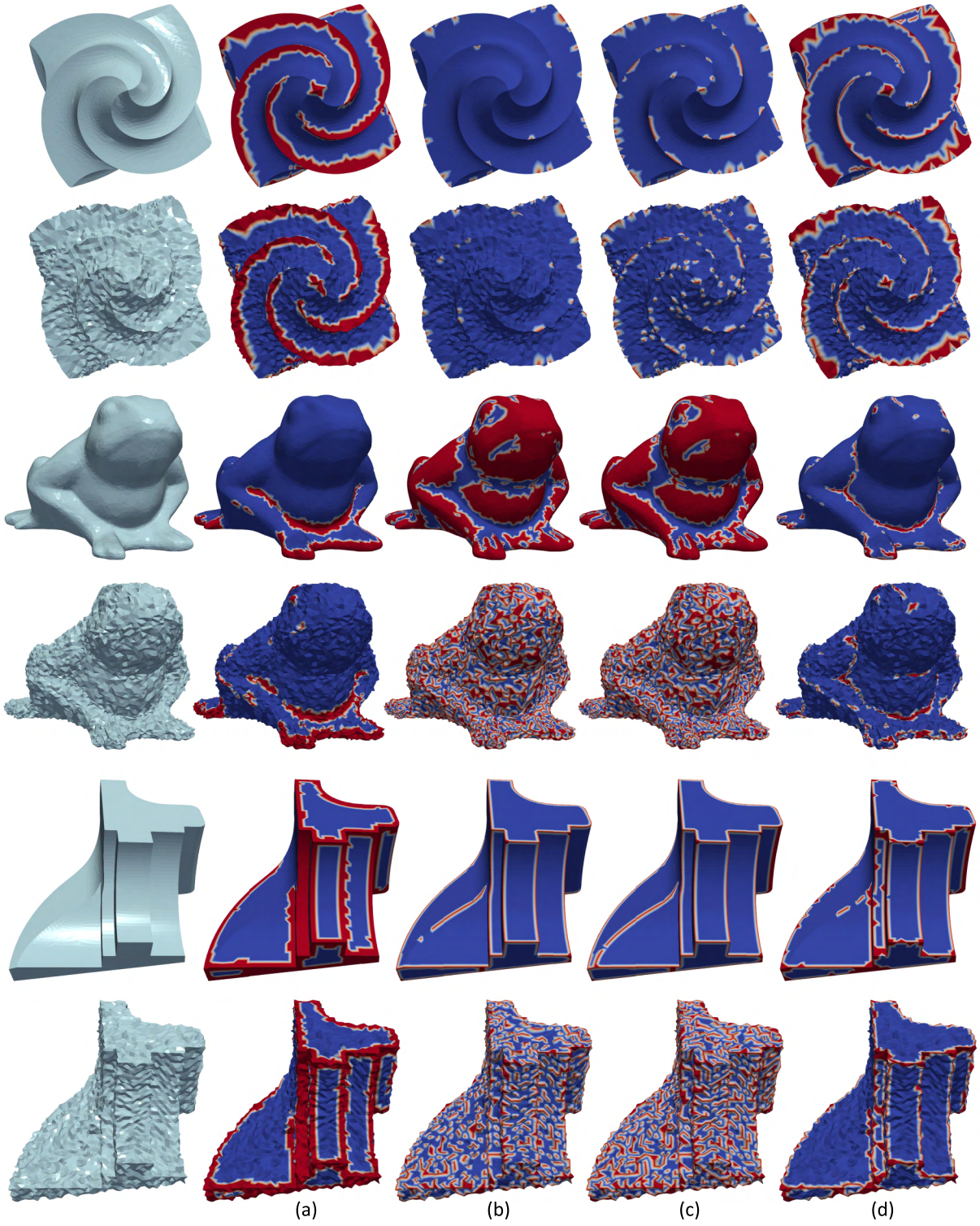


FIGURE 15. Feature detection of original and noisy 3D objects (Octa-flower, Frog, Fandisk) using different approaches. Specifically: (a) The method presented in [16], (b) l_1 based feature identification, (c) the method in [85] using the Eq. (55), (d) RPCA based feature identification.

applied in: (i) noisy objects, (ii) smoothed objects without intense features, pointing the need for an appropriate pre-processing step. The most robust method seems to be the RPCA based approach providing accurate results in any case.

FEATURE-AWARE 3D MESH DENOISING

The identification of features is a vital pre-processing step for many recent efficient denoising processes. Arvanitis *et al.* [16] classified each face as feature or

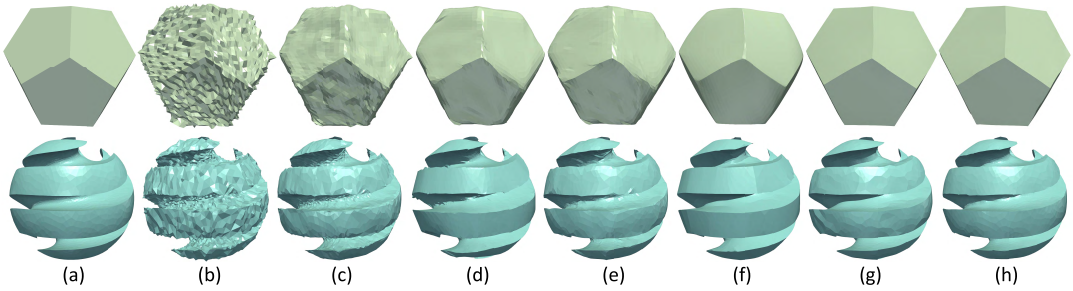


FIGURE 16. (a) Original models, (b) Noisy models and reconstructed denoising models of (c) Non Iterative [13], (d) Fast & Effective [19], (e) Bilateral Normal [14], (f) l_0 min [46], (g) Guided Normal Bilateral [44], (h) Feature-aware denoising [16].

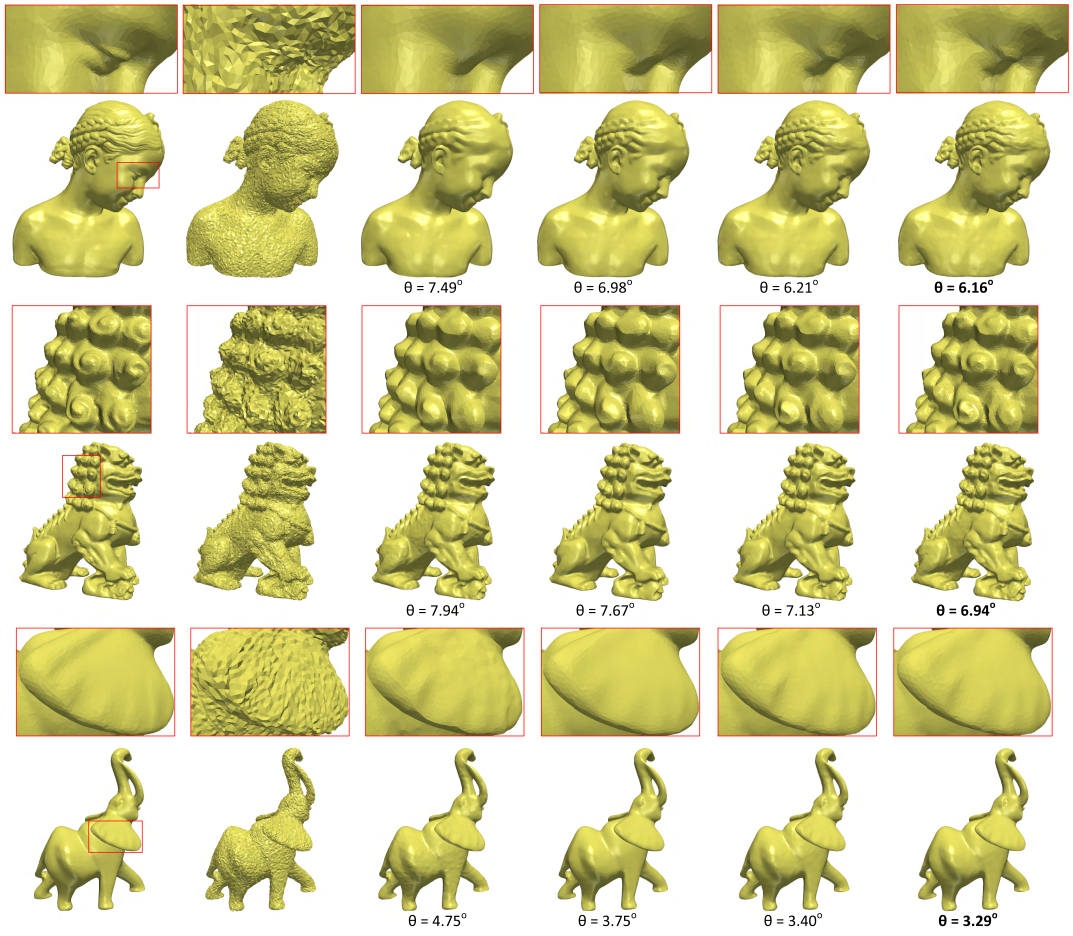


FIGURE 17. (a) Original 3D model, (b) Noisy model, (c) Guided Normals Bilateral [44], (d) Cascaded Normal Regression [47], (e) Parameter-free feature-aware denoising [16], (f) Feature-aware denoising using ideal parameters [16].

non-feature, in order to accelerate the execution time of the proposed method, by applying additional iterations only to the faces that were classified as features, reducing significantly the total computational complexity. Other methods (e.g., [44]), search for the ideal representative area of each face in order to estimate the guided normals. The identified features are used for selecting the best representative area for each face, improving significantly the outcome of the bilateral reconstruction process. Lu *et al.* [85] preserve features during the denoising process by performing a series of operations including feature detection, identification, and connection. To achieve the desired results, the authors propose a new feature-aware vertex update algorithm that takes advantage of local geometric neighboring information of each vertex and updates vertices in a feature-aware manner. In the rest part of this subsection, we provide evaluation studies of different feature aware denoising approaches as compared to other state-of-the-art methods for reconstructing various noisy 3D models. In Figures 16, 17 we show the reconstruction results, while the reconstruction

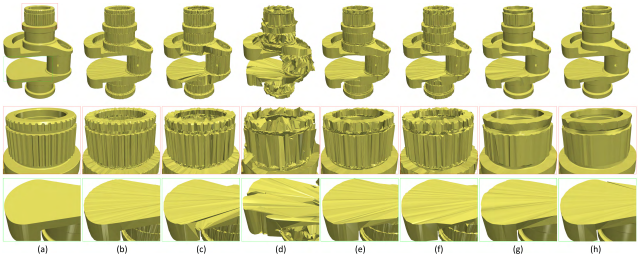


FIGURE 18. (a) Original mesh, (b) noisy mesh, (c) Fleisman *et al.* [12], (d) Sun *et al.* [19], (e) Zheng *et al.* (local), (f) Zheng *et al.* (global) [14], (g) Zhang *et al.* [44], (h) Arvanitis *et al.* [16].

quality is evaluated using the average angle normal difference θ.

Although the aforementioned approaches are computationally efficient and work well for various noisy inputs, potential users have to fine-tune the parameters and iteration numbers to obtain satisfactory results for meshes with different geometry features and noise levels. Motivated by this observation, we decided to include also results (see Fig. 17) using data-driven approaches (e.g., [16], [47]), that formulate the denoising process with cascaded non-linear regression functions and learn them from a set of noisy meshes and their ground-truth counterparts.

These methods, use training sets of noisy objects, scanned by the same devices, and they are appropriate for removing noise with similar characteristics. While many geometric features are reconstructed adequately, geometric details are not always preserved. In other words, the data-driven methods require a large dataset of models, transforming training into a very time-consuming process.

In Tables 2-3 we provide some additional de-noising results, using as evaluation metrics both the Hausdorff distance error and the average normal angle difference θ.

The presented evaluation studies help us identify several limitations that appear when considering complex noise patterns and/or complex geometries. More specifically, the effect of noise in the estimated bilateral parameters affects also the reconstruction accuracy of the geometric features, e.g., [14], [19], [44]. Though it should be noted that the aforementioned methods can be successfully applied for identifying sharp features and not medium or small-scale features. The method in [83] is an iterative approach combining pre-filtering, feature detection, and l_1-based feature recovery. The identified limitations of the presented approach is the increased computational complexity and the fact that in many cases the identified sharp features are irregularly highlighted. The l_0 presented in [46], is robust to noise with Gaussian

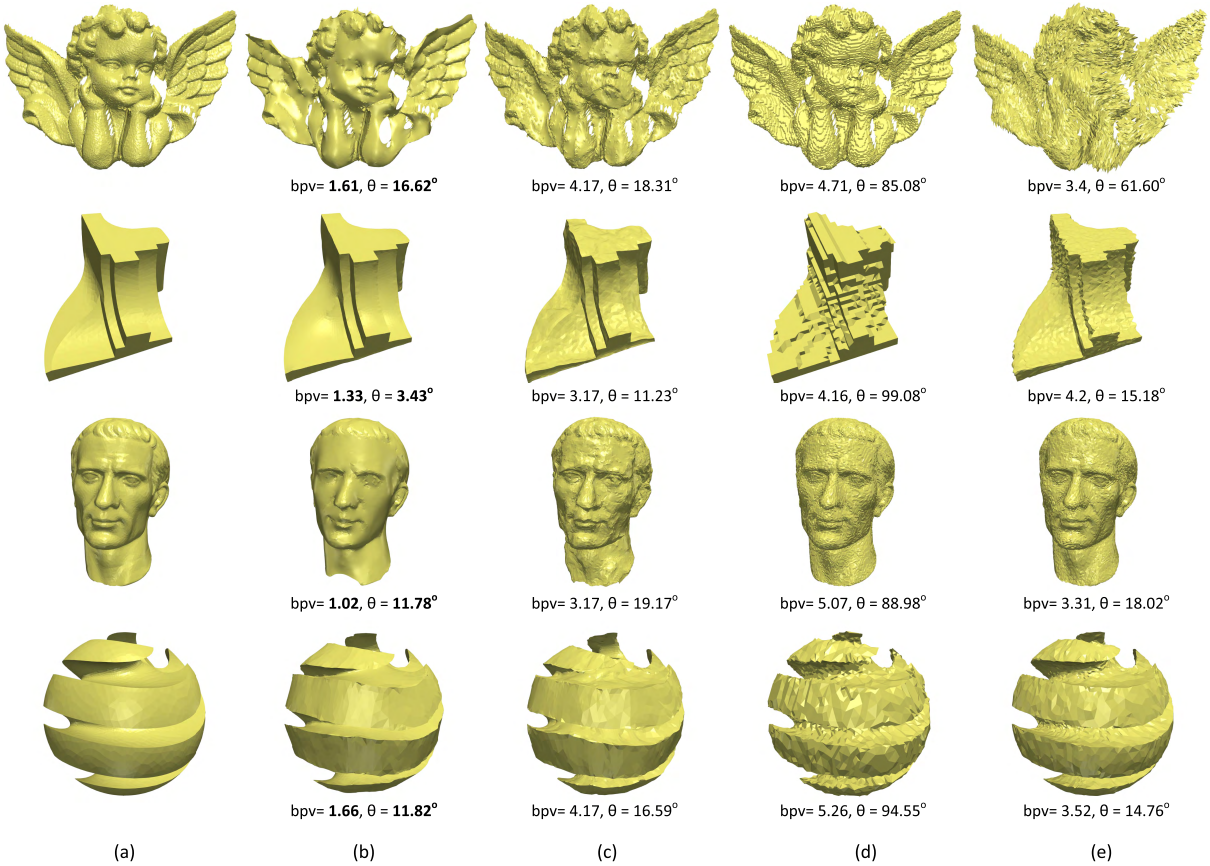


FIGURE 19. (a) Original Models vs Reconstruction results using: (b) feature-aware compression approach using Algorithm 3, (c) Uniform high pass quantization of different 3D models, (d) The OD3GC approach of [66] and, (e) Spectral compression in parts (∼ 600 vertices per part).

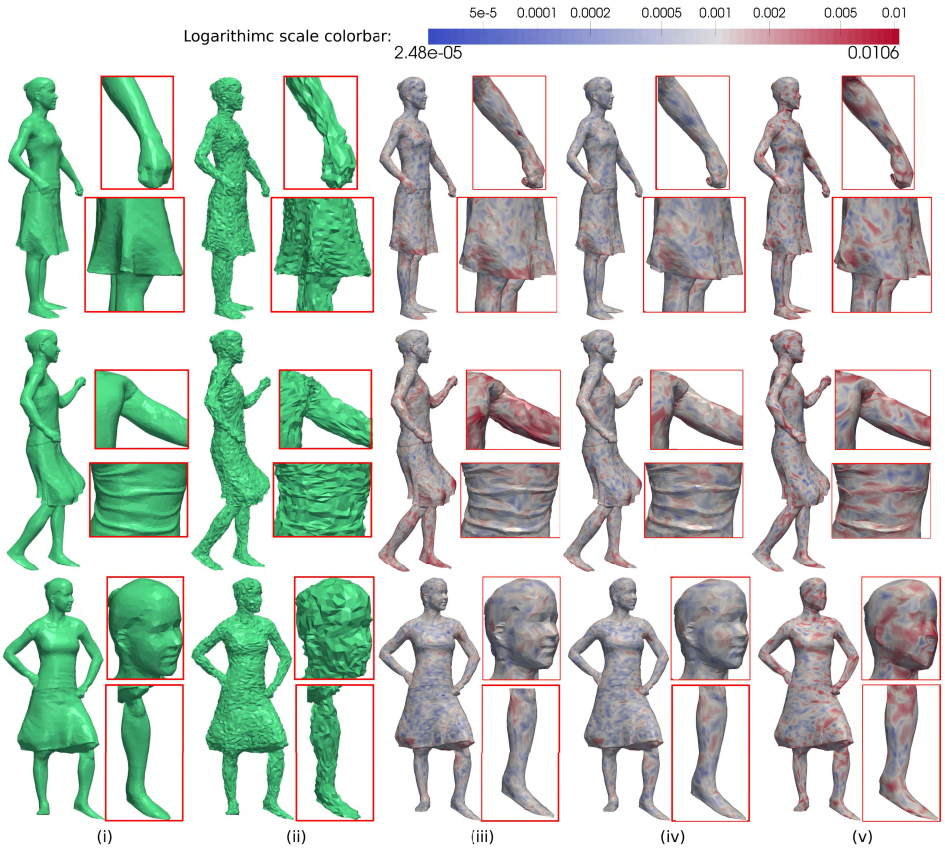


FIGURE 20. Reconstruction results using Samba model, (i) Original mesh, (ii) Noisy mesh, (iii) RPCA applied to the guided normals selected of the optimal overlapped patches, (iv) RPCA applied to the normals, (v) Bilateral normal filtering approach applied to each frame individually.

distribution, however, its performance is significantly reduced in more complex noise patterns (e.g., anisotropic noise, real-scan noise). Despite the good surface reconstruction results they cannot always preserve sharp features well, while in many cases their complexity becomes prohibitive. Finally, the presented denoising approaches cannot be used for reconstructing noisy 3D models where poorly-shaped triangles are formed by vertices mostly lying on sharp edges, (please refer to Fig. 18), a problem that remains a challenge for the future.

FEATURE-AWARE 3D MESH COMPRESSION

The identified features can be used also for efficiently compressing the geometry of a 3D mesh. To be more specific, features convey important visual information that should be preserved in lossy compression approaches in order to provide reconstruction output where the error is not easily perceived. Lalos *et al.* [84] suggest quantizing the δ coordinates of the geometric features and assigning zeros to rest delta coordinates that correspond to non feature points. Thus the encoding vector is written as $\boldsymbol{\delta}_z = [\delta_{z_1}, \ldots, \delta_{z_n}]$, where, $\delta_{z_i} = Q(\delta_i)$, if v_i is feature and 0, otherwise, and $Q(\cdot)$ is a scalar quantization function. In addition to $\boldsymbol{\delta}_z$, Lalos *et al.* [84] also quantize a set of known control points, which are uniformly distributed on the model surface $\mathbf{v}_c = Q([\mathbf{v}_{i_1}, \ldots, \mathbf{v}_{i_a}])$ where i_a is the vertex index and a correspond to the 1% of the total number of vertices n. The reconstruction of the 3D mesh vertices is performed as in [33], by solving a sparse linear system:

$$\begin{bmatrix} \mathbf{L} \\ \mathbf{I}_a \end{bmatrix} \mathbf{v} = \begin{bmatrix} \boldsymbol{\delta}_z \\ \mathbf{I}_a \mathbf{v}_c \end{bmatrix} \tag{57}$$

where $\mathbf{L}$ is the Laplacian operator defined in Eq. (5). The proposed compression approach is shortly presented in Algorithm 3.

Fig. 19 shows the reconstructed meshes at the decoder for different compression ratios using the following approaches: 1) a GFT based compression scheme (Spectral approach) [21], [24] [2] 2) the uniform high pass quantization approach presented in [33] and the 3) the OD3GC method [66] that performs direct quantization to the 3D-space coordinates. By inspecting these figure, it is obvious that the feature aware compression approach, enables aggressive compression ratios, that correspond to 0.5 bit per vertex (bpv), without introducing significant loss on the visual quality, measured

[2]Note that spectral method is executed in overlapped parts as described in [21] due to complexity constraints

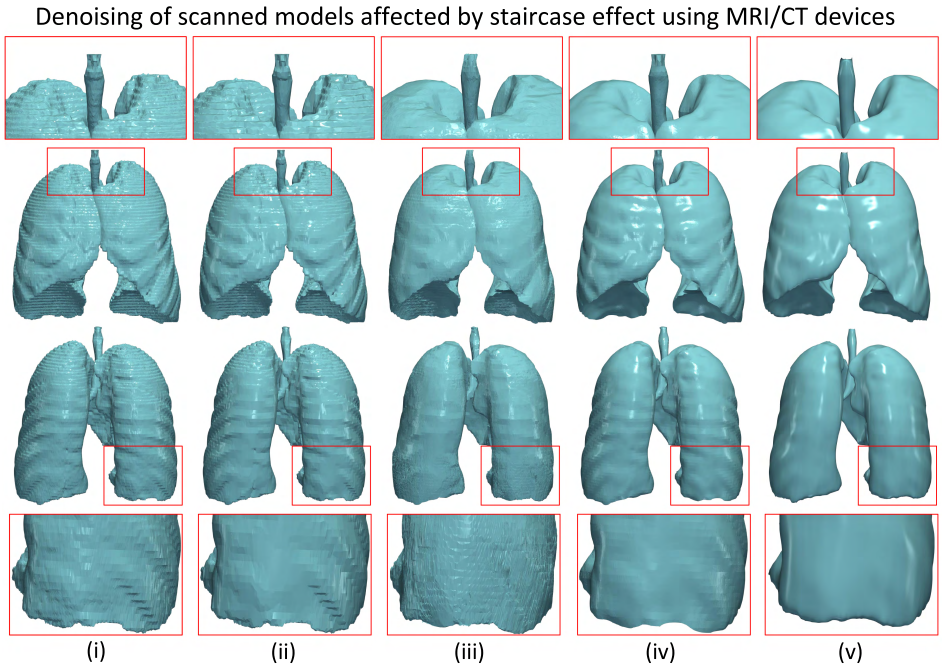


FIGURE 21. Triangulated details of lung model affected by the staircase effect (i) noisy mesh, and denoising result: (ii) l_0 min [46], (iii) bilateral normal [14], (iv) feature-aware denoising [16], (v) staircase-aware smoothing [63].

Algorithm 3 Feature-Aware High Pass Quantization

Require: Model $\mathbf{M}$
Ensure: Decompressed model $\tilde{\mathbf{M}}$
for $f = 1, \ldots,$ NumberofFrames **do**
Construct a matrix with Guided Normals, as shown in Fig. 12;
Evaluate features by executing RPCA as described in Eqs. (25)-(28).
Evaluate delta coordinates of feature points using Eq. (2)
Evaluate delta coordinates of a sparse set of anchors and perform quantization
At the decoder site solve a sparse linear system, see Eq. (57)
end for

Algorithm 4 Denoising of Dynamic Meshes Using RPCA

Require: Noisy animation $\mathbf{M}$
Ensure: Denoised animation $\hat{\mathbf{M}}$
for $f = 1, \ldots,$ NumberofFrames **do**
Estimate the face normals for each frame mesh $\mathbf{M}_f$ individually via Eq. (3)
Construction of a low-rank Matrix $\mathbf{M}_n$ according to Fig. 13
Execute RPCA on normals via Eq. (25)-(26) and estimate $\mathbf{P}_n, \mathbf{S}_n$
if $\mathbf{S}_{f_i} <$ threshold **then**
$\mathbf{n}_{f_i} \leftarrow \mathbf{P}_{f_i}$;
else
$\mathbf{n}_{f_i} \leftarrow \mathbf{P}_{f_i} + \mathbf{S}_{f_i}$;
end if
end for
for t = 1,...,NumberofIterations **do**
$\hat{\mathbf{v}}_{i_j}^{(t+1)} = \hat{\mathbf{v}}_{i_j}^{(t)} + \frac{1}{|\mathcal{F}_{i_j}|} \sum_{z \in \mathcal{F}_{i_j}} \hat{\mathbf{n}}_z \left[\hat{\mathbf{n}}_z^T \left(\hat{\mathbf{c}}_i^{(t)} - \hat{\mathbf{v}}_{i_j}^{(t)} \right) \right]$;
$\hat{\mathbf{c}}_i^{(t)} = \left(\hat{\mathbf{v}}_{i_1}^{(t)} + \hat{\mathbf{v}}_{i_2}^{(t)} + \hat{\mathbf{v}}_{i_3}^{(t)} \right) /3$;
end for

using the NMSVE. Finally, feature agnostic approaches introduce high-frequency errors affecting significantly the appearance of the surface, resulting in a blocky structure, where the errors are highly noticeable even in compression ratios which are higher than 2.5 bpv.

RPCA-BASED DYNAMIC MESH DENOISING

A similar approach can be also used for dynamic mesh compression by applying RPCA to a matrix $\mathbf{M}_c \in \Re^{3n_f \times \kappa k}$ where κ is the number of faces in the geometrical or topological neighborhood and k is the number of frames. Guided normals resulted in very smooth reconstructed surfaces. To address, this limitation we suggest executing RPCA directly to a matrix with normals. The robustness, in that case, is ensured by the exploitation of the temporal coherence (refer to Fig. 13). After identifying the geometric features in each frame, we evaluate the reconstructed vertices by executing the iterative process defined in Eqs. (12), (13). The proposed geometry reconstruction algorithm is summarized in Algorithm 4. By inspecting the dynamic mesh denoising

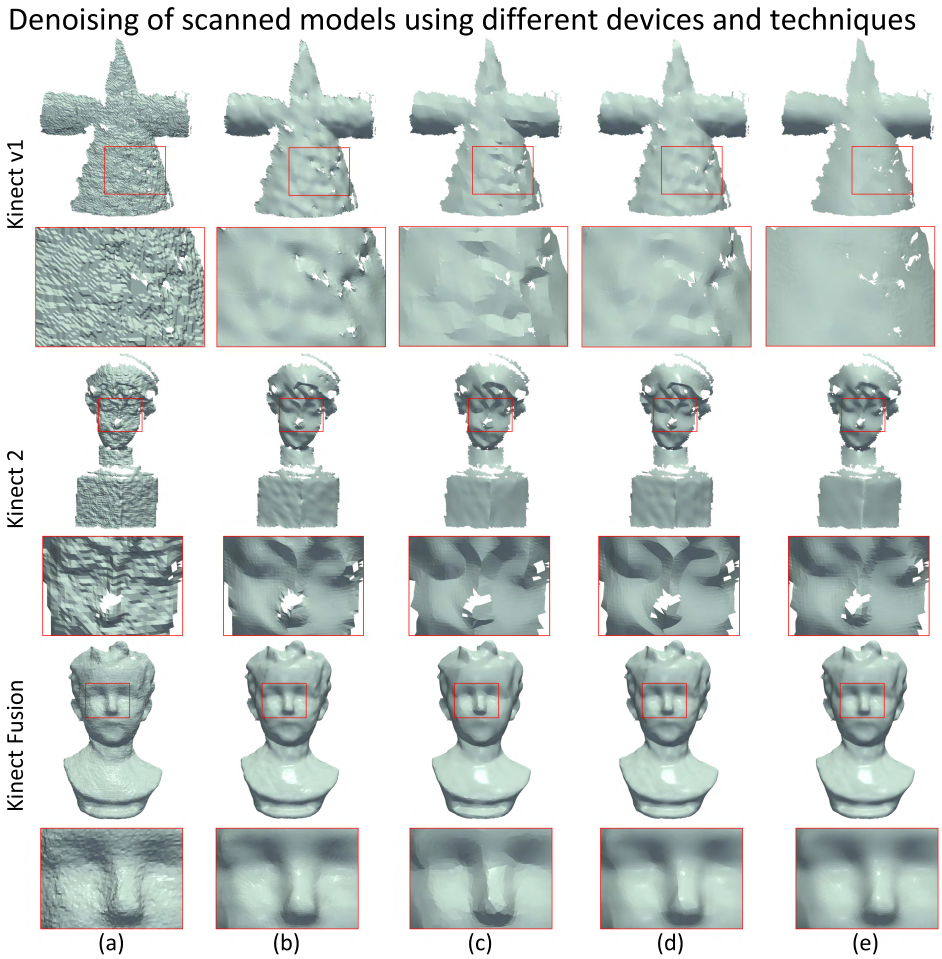


FIGURE 22. Denoising of real scanned 3D models (cone, girl, boy): (a) noisy, (b) bilateral normal [14], (c) l_0 min [46], (d) bilateral guided normal [44], (e) feature-aware denoising [16].

results in Fig. 20, it is clearly shown that the use of the individual normals, preserves more geometric details than the used of guided normals.

V. SUMMARY AND OPEN ISSUES

This article presents a comprehensive review of new methodologies that cross-fertilize sparsity-aware signal processing tools with digital geometry processing. The analyzed approaches support fast and efficient outlier rejection, denoising and completion of fully dynamic scenarios with an undefined motion pattern and complex topology modification behavior. In the heart of the reviewed approaches lie several sparse modeling and optimization tools, including Compressed Sensing (CS), Matrix Completion (MC) and Robust PCA (RPCA) approaches simultaneously maintaining *fast processing times*, *plausible reconstruction results* and *out-of-core behavior*. In Figs. 21-22 we illustrate the impact of the presented signal processing tools on a heritage science and a medical example where we process: i) sculpture models scanned by off the shelf devices (e.g., Kinect v1, v2) (see Fig. 22) and 3D lung models reconstructed from CTs/MRIs (see Fig. 21). Although the presented approaches are expected to play a crucial role in this exciting endeavor, there are a lot of open issues awaiting investigation.

The scale of acquired data in real time 3D scanning operations is growing very quickly, requiring the accurate reconstruction of entire scenes with various objects represented by structured shapes. These requirements introduce significant scientific challenges, facilitating a wide number of applications with tight timing restrictions, such as tele-immersion, real-time surface mapping, and tracking, aero-reconstruction for disaster management. Therefore in order to support such challenging problems, the presented approaches should be able to work in a real-time fashion by utilizing both adaptive and distributed approaches.

Another important application of the presented sparse optimization techniques is the consolidation of dynamic point clouds, in order to enhance the captured data. We have seen that graph-based matrix completion techniques can successfully recover the unknown points with only 30% of the original data. The optimization techniques are based on the sparsity of the matrix in the singular value domain as well

as the exploitation of the spatial coherence. The performance of the techniques is depending on the parameters which give different weights to each domain, depending on the specific model to be reconstructed. Therefore, techniques that obtain the values of these parameters in an optimal manner for each specific model would be an important extension of the presented techniques. Moreover, when the whole point cloud is available for processing, a hierarchical or multi-layer approach for the recovery of the missing points could improve even more the performance of the techniques.

Despite the tremendous progress on the landscape of the 3D mesh processing field, we believe that our approaches provides a novel insight at a key area with renewed research interest, where high potential for novel improvements such as online clustering and processing based on the motion, the number and the scale of geometric features, is feasible in the near future.

REFERENCES

[1] F. Bernardini, H. Rushmeier, I. M. Martin, J. Mittleman, and G. Taubin, "Building a digital model of Michelangelo's Florentine Pieta," *IEEE Comput. Graph. Appl.*, vol. 22, no. 1, pp. 59–67, Jan. 2002.

[2] F. Bernardini, H. Rushmeier, I. M. Martin, J. Mittleman, and G. Taubin, "Building a digital model of Michelangelo's Florentine Pieta," *IEEE Comput. Graph. Appl.*, vol. 22, no. 1, pp. 59–67, Jan./Feb. 2002, doi: 10.1109/38.974519.

[3] M. Berger *et al.*, "State of the art in surface reconstruction from point clouds," in *Proc. EUROGRAPHICS Star Rep.*, vol. 1, no. 1, 2014, pp. 161–185.

[4] Y. Wang, L. Wang, W. Hao, X. Ning, Z. Shi, and M. Zhao, "A novel slicing-based regularization method for raw point clouds in visible IoT," *IEEE Access*, vol. 6, pp. 18299–18309, 2018.

[5] A. Anwer, S. S. A. Ali, A. Khan, and F. Mériaudeau, "Underwater 3-D scene reconstruction using Kinect v2 based on physical models for refraction and time of flight correction," *IEEE Access*, vol. 5, pp. 15960–15970, 2017.

[6] L. Pekař and Q. Gao, "Spectrum analysis of LTI continuous-time systems with constant delays: A literature overview of some recent results," *IEEE Access*, vol. 6, pp. 35457–35491, 2018.

[7] Y. Zhong and F. Chen, "Computing medial axis transformations of 2D point clouds," *Graph. Models*, vol. 97, pp. 50–63, May 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1524070318300092

[8] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon Mesh Processing*. Boca Raton, FL, USA: CRC Press, 2010.

[9] O. Sorkine, "Differential representations for mesh processing," *Comput. Graph. Forum*, vol. 25, no. 4, pp. 789–807, 2006.

[10] G. Taubin, "A signal processing approach to fair surface design," in *Proc. 22nd Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, New York, NY, USA, 1995, pp. 351–358, doi: 10.1145/218380.218473.

[11] N. Himayat and S. A. Kassam, "Approximate performance analysis of edge preserving filters," *IEEE Trans. Signal Process.*, vol. 41, no. 9, pp. 2764–2777, Sep. 1993.

[12] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 950–953, 2003.

[13] T. R. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 943–949, 2003.

[14] Y. Zheng, H. Fu, O. K.-C. Au, and C.-L. Tai, "Bilateral normal filtering for mesh denoising," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 10, pp. 1521–1530, Oct. 2013.

[15] J. Solomon, K. Crane, A. Butscher, and C. Wojtan. (2014). "A general framework for bilateral and mean shift filtering." [Online]. Available: https://arxiv.org/abs/1405.4734

[16] G. Arvanitis, A. Lalos, K. Moustakas, and N. Fakotakis, "Feature preserving mesh denoising based on graph spectral processing," *IEEE Trans. Vis. Comput. Graphics*, to be published, doi: 10.1109/TVCG.2018.2802926.

[17] W. T. Tutte, "How to draw a graph," *Proc. London Math. Soc.*, vol. s3-13, no. 1, pp. 743–767, 1963.

[18] E. BrianDavies, G. M. L. Gladwell, J. Leydold, and P. F. Stadler, "Discrete nodal domain theorems," *Linear Algebra Appl.*, vol. 336, nos. 1–3, pp. 51–60, 2001.

[19] X. Sun, P. Rosin, R. Martin, and F. Langbein, "Fast and effective feature-preserving mesh denoising," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 5, pp. 925–938, Sep. 2007.

[20] L. Váša, S. Marras, K. Hormann, and G. Brunnett, "Compressing dynamic meshes with geometric Laplacians," *Comput. Graph. Forum*, vol. 33, no. 2, pp. 145–154, 2014.

[21] A. S. Lalos, I. Nikolas, E. Vlachos, and K. Moustakas, "Compressed sensing for efficient encoding of dense 3D meshes using model-based Bayesian learning," *IEEE Trans. Multimedia*, vol. 19, no. 1, pp. 41–53, Jan. 2017.

[22] A. S. Lalos, A. A. Vasilakis, A. Dimas, and K. Moustakas, "Adaptive compression of animated meshes by exploiting orthogonal iterations," *Vis. Comput.*, vol. 33, nos. 6–8, pp. 811–821, 2017, doi: 10.1007/s00371-017-1395-4.

[23] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 97:1–97:9, Aug. 2008, doi: 10.1145/1360612.1360696.

[24] Z. Karni and C. Gotsman, "Compression of soft-body animation sequences," *Comput. Graph.*, vol. 28, no. 1, pp. 25–34, 2004.

[25] E. Vlachos, A. S. Lalos, A. Spathis-Papadiotis, and K. Moustakas, "Distributed consolidation of highly incomplete dynamic point clouds based on rank minimization," *IEEE Trans. Multimedia*, vol. 20, no. 12, pp. 3276–3288, Dec. 2018.

[26] J. Mairal, M. Elad, and F. Bach, "Guest editorial: Sparse coding," *Int. J. Comput. Vis.*, vol. 114, nos. 2–3, pp. 89–90, Sep. 2015, doi: 10.1007/s11263-015-0845-6.

[27] A. Qayyum *et al.*, "Image classification based on sparse-coded features using sparse coding technique for aerial imagery: A hybrid dictionary approach," *Neural Comput. Appl.*, Dec. 2017. [Online]. Available: https://link.springer.com/article/10.1007/s00521-017-3300-5#citeas, doi: 10.1007/s00521-017-3300-5.

[28] Y.-J. Yoon, A. Lelidis, A. C. Öztireli, J.-M. Hwang, M. Gross, and S.-M. Choi, "Geometry representations with unsupervised feature learning," in *Proc. Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2016, pp. 137–142.

[29] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or, "Consolidation of unorganized point clouds for surface reconstruction," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 176:1–176:7, Dec. 2009, doi: 10.1145/1618452.1618522.

[30] H. Avron, A. Sharf, C. Greif, and D. Cohen-Or, "ℓ_1-sparse reconstruction of sharp point set surfaces," *ACM Trans. Graph.*, vol. 29, no. 5, pp. 135:1–135:12, Nov. 2010, doi: 10.1145/1857907.1857911.

[31] E. Vlachos, A. S. Lalos, K. Moustakas, and K. Berberidis, "Efficient graph-based matrix completion on incomplete animated models," in *Proc. IEEE Int. Conf. Multimedia Expo*, Hong Kong, Jul. 2017, pp. 1121–1124.

[32] B. Kim and J. Rossignac, "GeoFilter: Geometric selection of mesh filter parameters," *Comput. Graph. Forum*, vol. 24, no. 3, pp. 295–302, 2005.

[33] O. Sorkine, "Laplacian mesh processing," in *Proc. Eurographics-State Art Reports (STARs)*, 2005, pp. 53–70.

[34] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Computing and rendering point set surfaces," *IEEE Trans. Vis. Comput. Graphics*, vol. 9, no. 1, pp. 3–15, Jan. 2003, doi: 10.1109/TVCG.2003.1175093.

[35] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, "Parameterization-free projection for geometry reconstruction," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007, Art. no. 22, doi: 10.1145/1276377.1276405.

[36] S. Liu, K.-C. Chan, and C. C. L. Wang, "Iterative consolidation of unorganized point clouds," *IEEE Comput. Graph. Appl.*, vol. 32, no. 3, pp. 70–83, May/Jun. 2012, doi: 10.1109/MCG.2011.14.

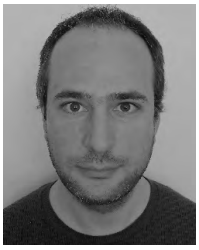
[37] J. Wang *et al.*, "Consolidation of low-quality point clouds from outdoor scenes," in *Proc. 11th Eurographics/ACMSIGGRAPH Symp. Geometry Process.*, 2013, pp. 207–216, doi: 10.1111/cgf.12187.

[38] G. Arvanitis, A. S. Lalos, K. Moustakas, and N. Fakotakis, "Weighted regularized laplacian interpolation for consolidation of highly-incomplete time varying point clouds," in *Proc. 3DTV Conf., True Vis.-Capture, Transmiss. Display 3D Video (3DTV-CON)*, Jun. 2017, pp. 1–4.

[39] G. Arvanitis, A. Spathis-Papadiotis, A. S. Lalos, K. Moustakas, and N. Fakotakis, "Outliers removal and consolidation of dynamic point cloud," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3888–3892.
[40] M. Zhong and H. Qin, "Surface inpainting with sparsity constraints," *Comput. Aided Geometric Des.*, vol. 41, pp. 23–35, Jan. 2016.
[41] G. Arvanitis, A. S. Lalos, K. Moustakas, and N. Fakotakis, "Real-time removing of outliers and noise in 3D point clouds applied in robotic applications," in *Interactive Collaborative Robotics*, A. Ronzhin, G. Rigoll, and R. Meshcheryakov, Eds. Cham, Switzerland: Springer, 2017, pp. 11–19.
[42] K. Hildebrandt and K. Polthier, "Anisotropic filtering of non-linear surface features," *Comput. Graph. Forum*, vol. 23, no. 3, pp. 391–400, 2004.
[43] C. L. Bajaj and G. Xu, "Anisotropic diffusion of surfaces and functions on surfaces," *ACM Trans. Graph.*, vol. 22, no. 1, pp. 4–32, Jan. 2003.
[44] W. Zhang, B. Deng, J. Zhang, S. Bouaziz, and L. Liu, "Guided mesh normal filtering," *Comput. Graph. Forum*, vol. 34, no. 7, pp. 23–34, 2015.
[45] R. Wang, Z. Yang, L. Liu, J. Deng, and F. Chen, "Decoupling noise and features via weighted ℓ_1-analysis compressed sensing," *ACM Trans. Graph.*, vol. 33, no. 2, pp. 18:1–18:12, Apr. 2014, doi: 10.1145/2557449.
[46] L. He and S. Schaefer, "Mesh denoising via L0 minimization," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 64:1–64:8, Jul. 2013, doi: 10.1145/2461912.2461965.
[47] P.-S. Wang, Y. Liu, and X. Tong, "Mesh denoising via cascaded normal regression," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 232:1–232:12, Nov. 2016.
[48] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3D point cloud sequences," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1765–1778, Apr. 2016.
[49] A. S. Lalos, I. Nikolas, and K. Moustakas, "Sparse coding of dense 3D meshes in mobile cloud applications," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Dec. 2015, pp. 403–408.
[50] K. Wolff *et al.*, "Point cloud noise and outlier removal for image-based 3D reconstruction," in *Proc. IEEE 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 118–127.
[51] B. M. Brown, "Statistical uses of the spatial median," *J. Roy. Stat. Soc. B, Methodol.*, vol. 45, no. 1, pp. 25–30, 1983.
[52] E. Mattei and A. Castrodad, "Point cloud denoising via moving RPCA," *Comput. Graph. Forum*, vol. 36, no. 8, pp. 123–137, 2017, doi: 10.1111/cgf.13068.
[53] N. A. Campbell, "Robust procedures in multivariate analysis I: Robust covariance estimation," *J. Roy. Stat. Soc. C, Appl. Statist.*, vol. 29, no. 3, pp. 231–237, 1980.
[54] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, p. 11, May 2011.
[55] E. Bayram, P. Frossard, E. Vural, and A. Alatan, "Analysis of airborne LiDAR point clouds with spectral graph filtering," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 8, pp. 1284–1288, Aug. 2018.
[56] D. T. Ngo, J. Östlund, and P. Fua, "Template-based monocular 3D shape recovery using Laplacian meshes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 172–187, Jan. 2016.
[57] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-ICP: A globally optimal solution to 3D ICP point-set registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016.
[58] A. P. Bustos and T.-J. Chin, "Guaranteed outlier removal for point cloud registration with correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2868–2882, Dec. 2018.
[59] A. W. Vieira, P. L. J. Drews, and M. F. M. Campos, "Spatial density patterns for efficient change detection in 3D environment for autonomous surveillance robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 766–774, Jul. 2014.
[60] S. Xiong, J. Zhang, J. Zheng, J. Cai, and L. Liu, "Robust surface reconstruction via dictionary learning," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 201:1–201:12, Nov. 2014, doi: 10.1145/2661229.2661263.
[61] B. Li, W. Jiang, Z. Cheng, G. Dang, and S. Jin, "Feature preserving consolidation for unorganized point clouds," in *Proc. IEEE Int. Conf. Prog. Inform. Comput.*, vol. 2, Dec. 2010, pp. 892–895.
[62] X. Wang, X. Liu, and H. Qin, "Robust surface consolidation of scanned thick point clouds," in *Proc. Int. Conf. Comput.-Aided Design Comput. Graph.*, Nov. 2013, pp. 38–43.
[63] T. Moench, S. Adler, and B. Preim, "Staircase-aware smoothing of medical surface meshes," in *Proc. Eurograph. Workshop Vis. Comput. Biol. Med.*, D. Bartz, C. Botha, J. Hornegger, R. Machiraju, A. Wiebel, and B. Preim, Eds. The Eurographics Association, 2010, doi: 10.2312/VCBM/VCBM10/083-090.
[64] X. Bingqian, G. Yanfeng, and C. Zhimin, "Building LiDAR point cloud denoising processing through sparse representation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 585–588.
[65] K. Sarkar, F. Bernard, K. Varanasi, C. Theobalt, and D. Stricker, "Structured low-rank matrix factorization for point-cloud denoising," in *Proc. Int. Conf. 3D Vis. (3DV)*, 2018, pp. 444–453.
[66] K. Mamou, T. Zaharia, and F. Prêteux, "TFAN: A low complexity 3D mesh compression algorithm," *Comput. Animation Virtual Worlds*, vol. 20, nos. 2–3, pp. 343–354, Jun. 2009.
[67] J. Castorena, C. D. Creusere, and D. Voelz, "Modeling lidar scene sparsity using compressive sensing," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2010, pp. 2186–2189.
[68] J. Hou, L.-P. Chau, M. Zhang, N. Magnenat-Thalmann, and Y. He, "A highly efficient compression framework for time-varying 3-D facial expressions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 9, pp. 1541–1553, Sep. 2014.
[69] J. Hou, L.-P. Chau, N. Magnenat-Thalmann, and Y. He, "Sparse low-rank matrix approximation for data compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 5, pp. 1043–1054, May 2017.
[70] D. Banerjee, K. Yu, and G. Aggarwal, "Robotic arm based 3D reconstruction test automation," *IEEE Access*, vol. 6, pp. 7206–7213, 2018.
[71] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and optimization for big data analytics: (Statistical) learning tools for our era of data deluge," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 18–31, Sep. 2014.
[72] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM J. Optim.*, vol. 21, no. 2, pp. 572–596, 2011.
[73] B. Vallet, M. Brédif, A. Serna, B. Marcotegui, and N. Paparoditis, "TerraMobilita/iQmulus urban point cloud analysis benchmark," *Comput. Graph.*, vol. 49, pp. 126–133, Jun. 2015. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01167995
[74] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, p. 717, 2009, doi: 10.1007/s10208-009-9045-5.
[75] Y. Deng, Q. Dai, and Z. Zhang, "Graph Laplace for occluded face completion and recognition," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2329–2338, Aug. 2011.
[76] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.
[77] G. Arvanitis, K. Moustakas, N. Fakotakis, and A. S. Lalos, "3D mesh inpainting using matrix completion via augmented Lagrange multiplier method," in *Proc. IEEE 13th Image, Video, Multidimensional Signal Process. Workshop (IVMSP)*, Jun. 2018, pp. 1–5.
[78] Y. Deng, Y. Liu, Q. Dai, Z. Zhang, and Y. Wang, "Noisy depth maps fusion for multiview stereo via matrix completion," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 5, pp. 566–582, Sep. 2012.
[79] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011, doi: 10.1561/2200000016.
[80] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, 2002, pp. 1–130.
[81] O. Sorkine and D. Cohen-Or, "Least-squares meshes," in *Proc. Shape Modeling Appl.*, Jun. 2004, pp. 191–199.
[82] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[83] R. Wang, Z. Yang, L. Liu, J. Deng, and F. Chen, "Decoupling noise and features via weighted ℓ_1-analysis compressed sensing," *ACM Trans. Graph.*, vol. 33, no. 2, pp. 18:1–18:12, 2014.
[84] A. S. Lalos, G. Arvanitis, A. Spathis-Papadiotis, and K. Moustakas, "Feature aware 3D mesh compression using robust principal component analysis," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.
[85] X. Lu, Z. Deng, and W. Chen, "A robust scheme for feature-preserving mesh denoising," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 3, pp. 1181–1194, Mar. 2016.

ARIS S. LALOS received the Diploma, M.A.Sc., and Ph.D. degrees from the Computer Engineering and Informatics Department (CEID), School of Engineering (SE), University of Patras (UoP), Patras, Greece, in 2003, 2005, and 2010, respectively. He was a Research Fellow with the Signal Processing and Communications Laboratory, CEID, SE, UoP, from 2005 to 2010, and the Signal Theory and Communications Department, Technical University of Catalonia, Barcelona, Spain, from 2012 to 2014. From 2011 to 2012, he was a Telecommunication Research Engineer with Analogies S.A, an early stage start-up. He has been a Research Fellow with the Visualization and Virtual Reality Group, since 2015. In 2018, he was an Elected Principal Researcher (Associate Research Professor Level with tenure) with the Industrial Systems Institute, ATHENA Research Centre. He has authored 74 research papers in international journals (27), conferences (44), and edited books (3). His general research interests include digital communications, adaptive filtering algorithms, geometry processing, wireless body area networks, and biomedical signal processing. He received the Best Demo Award at the IEEE CAMAD 2014, the Best Paper Award at the IEEE ISSPIT 2015, and the World's FIRST 10K Best Paper Award at the IEEE ICME 2017. In 2015, he was nominated as an Exemplary Reviewer for the IEEE COMMUNICATIONS LETTERS. He has participated in several European projects related to the ICT and eHealth domain (e.g., COOPCOM, ALPHA, WSN4QoL, KinOptim, MOMIRAS, and MyAirCoach). He acts as a regular reviewer for several technical journals.

EVANGELOS VLACHOS received the Diploma, M.Sc., and Ph.D. degrees from the Computer Engineering and Informatics Department, University of Patras, Greece, in 2005, 2009, and 2015, respectively. From 2015 to 2016, he was a Postdoctoral Researcher with the Signal Processing and Communications Laboratory, University of Patras, working on sparse signal processing for distributed networks, where he was also a Postdoctoral Researcher with the Visualization and Virtual Reality Group, working on graph-based optimization for reconstruction of 3D models, from 2016 to 2017. His research interests include sparse signal modeling tools, and distributed and graph-based optimization for static and time-varying point clouds. He received the Best Paper Award (World's First 10K) from the IEEE International Conference on Multimedia and Expo 2017.



GERASIMOS ARVANITIS received the M.Sc. degree in electronics and information processing from the University of Patras, Patras, Greece, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering Department. His main research interests include digital geometry processing, 3D modeling, reconstruction, feature preserving denoising algorithms, and computer graphics.

KONSTANTINOS MOUSTAKAS received the Diploma and Ph.D. degrees in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2003 and 2007, respectively. From 2007 to 2011, he was a Postdoctoral Research Fellow with the Centre for Research and Technology Hellas, Information Technologies Institute, Greece. He is currently an Associate Professor with the Electrical and Computer Engineering Department, University of Patras, Patras, Greece, where he is also the Head of the Visualization and Virtual Reality Group. He has authored or co-authored more than 120 papers in refereed journals, edited books, and international conferences. His main research interests include computer vision, computer graphics, computational geometry, and virtual physiological humans. He is an IEEE Senior Member and a member of the Eurographics Society. He has participated in more than 17 research and development projects funded by the EC and the Greek Secretariat of Research and Technology.

KOSTAS BERBERIDIS received the Diploma degree in electrical engineering from the Democritus University of Thrace, Komotini, Greece, in 1985, and the Ph.D. degree in signal processing and communications from the University of Patras, Patras, Greece, in 1990. In 1991, he was with the Signal Processing Laboratory, National Defense Research Center. From 1992 to 1994 and from 1996 to 1997, he was a Researcher with the Computer Technology Institute, Patras. From 1994 to 1995, he was a Postdoctoral Fellow with CCETT/CNET, Rennes, France. Since 1997, he has been with the Computer Engineering and Informatics Department, University of Patras, where he is currently a Professor and the Head of the Signal Processing and Communications Laboratory. Since 2008, he has been the Director of the Signal Processing and Communications Research Unit, Computer Technology Institute and Press Diophantus. His research interests include adaptive signal processing, distributed processing and learning, signal processing for communications, and wireless sensor networks. He has been serving as a member of the Board of Directors of EURASIP, since 2017. He is a member of the Signal Processing Theory and Methods Technical Committee of the IEEE Signal Processing Society and the Signal Processing for Communications and Electronics Technical Committee of the IEEE Communications Society. Since 2010, he has been serving as the Chair of the Greece Chapter of the IEEE Signal Processing Society. He has served or has been serving as a member of scientific and organizing committees of several international conferences, as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE SIGNAL PROCESSING LETTERS, and as the Guest Editor and the Associate Editor for the *EURASIP Journal on Advances in Signal Processing*.