Compressed Sensing for Efficient Encoding of Dense 3D Meshes Using Model-Based Bayesian Learning

Aris S. Lalos, *Member, IEEE*, Iason Nikolas, *Student Member, IEEE*, Evangelos Vlachos, *Member, IEEE*, and Konstantinos Moustakas, *Member, IEEE*

***Abstract*—With the growing demand for easy and reliable generation of 3D models representing real-world or synthetic objects, new schemes for acquisition, storage, and transmission of 3D meshes are required. In principle, 3D meshes consist of vertex positions and vertex connectivity. Vertex position encoders are much more resource demanding than connectivity encoders, stressing the need for novel geometry compression schemes. The design of an accurate and efficient geometry compression system can be achieved by increasing the compression ratio without affecting the visual quality of the object and minimizing the computational complexity. In this paper, we present novel compression/reconstruction schemes that enable aggressive compression ratios, without significantly reducing the visual quality. The encoding is performed by simply executing additions/subtractions. The benefits of the proposed method become more apparent as the density of the meshes increases, while it provides a flexible framework to trade efficiency for reconstruction quality. We derive a novel Bayesian learning algorithm that models the most significant graph Fourier transform coefficients of each submesh, as a multivariate Gaussian distribution. Then we evaluate iteratively the distribution parameters using the expectation-maximization approach. To improve the performance of the proposed approach in highly under determined problems, we exploit the local smoothness of the partitioned surfaces. Extensive evaluation studies, carried out using a large collection of different 3D models, show that the proposed schemes, as compared to the state-of-the-art approaches, achieve competitive compression ratios, offering at the same time significantly lower encoding complexity.**

I. Introduction

RECENTLY, there has been increasing interest from researchers, system designers, and application developers on acquiring, processing, transmitting and storing 3D models [1]–[3] facilitating several applications, e.g., mobile cloud gaming [4] and 3D Tele-immersion [5], [6]. In many of these applications, the reconstruction does not necessarily have to be exactly equal to the input data allowing some loss of precision that is not easily distinguishable. The exploitation of this property results in perception oriented compression schemes that degrade gracefully with aggressive Compression Ratios (CRs), opening up new possibilities for live acquisition, transmission and storage of dense 3D models.

The most common way of representing 3D models in applications is the polygon modeling that approximates surfaces using 3D meshes. In general, 3D meshes consist of vertices which provide the geometry information and polygons that connect the vertices and determine elementary surfaces. It is worth mentioning that the encoded geometry is on average more than five times larger than the encoded connectivity [7], since the raw geometry data, whether originating from scanned real-world objects or synthetic modeling applications [4], are represented using floating point precision. As a result, although state-of-the-art connectivity encoders are extremely effective [8], [9], the compression of geometry information not only seems to remain a challenge, [7] but also becomes more essential in mobile cloud computing settings [10] where the encoder (e.g., mobile device) transmission and processing resources are much more limited as compared to the decoder (e.g, cloud) processing resources.

A. Related Work and Contributions

Maglo *et al.* [11] have presented a survey paper, focusing on the latest developments on different mesh compression methods. The pioneering works of Deering [12], Taubin and Rossignac [13] as well as several following mesh-compression techniques, focus mainly on connectivity encoding (e.g., [14], [15]) ignoring the geometry information, that requires much more data for representation. Moreover, the authors in [9], showed that the efficiency of the connectivity encoding has reached near-optimal level and as a result motivated us to focus on the geometry compression and assume the vertex connectivity available at the decoder.

Traditional compression schemes [16], [17], suggest performing direct quantization to the three space coordinates. This form of quantization introduces high-frequency errors into the model that modify significantly the appearance of the surface, resulting in a 3D model with a blocky structure, where the reconstruction errors are highly noticeable. Hence, these algorithms are not suitable for "lossy" compression, due to their non-graceful degradation. To overcome this limitation, several works suggest to work in a different domain than the spatial one, where direct quantization will result in low frequency errors that are not easily distinguishable [18]–[20]. One of the first attempts in perception oriented mesh compression is high-pass encoding [18], which builds on the idea of quantizing the differential coordinates. This approach succeeds at capturing the local relation

Manuscript received March 12, 2016; revised June 9, 2016 and July 20, 2016; accepted August 24, 2016. Date of publication September 2, 2016; date of current version December 14, 2016. This work was supported by the Greek Secretariat for Research and Technology Bilateral Collaboration Project MOMIRAS (*ISR*_3215) and by the H2020-PHC-2014-2015 Project MyAirCoach under Grant 643607. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Cha Zhang.

A. S. Lalos, I. Nikolas, and K. Moustakas are with the Electrical and Computer Engineering Department, University of Patras, Patras 26504, Greece (e-mail: aris.lalos@ece.upatras.gr; iason.nikolas@ece.upatras.gr; moustakas@ece.upatras.gr).

E. Vlachos is with the Computer Engineering and Informatics Department, University of Patras, Patras 26504, Greece (e-mail: vlaxose@ceid.upatras.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2016.2605927

of vertices and usually outperforms the direct quantization approaches. Similarly, the authors in [19], [20] suggest performing compression by transmitting a small number of "low frequency" components estimated by projecting the Cartesian coordinates to the graph Fourier domain. The main drawback of the aforementioned schemes, is the increased processing demands at the transmitter, since they require the eigenvalue decomposition of the Laplacian matrix (e.g., evaluation of the graph Fourier basis vectors) and the projection of the vertex coordinates to a subspace, defined by these vectors.

To overcome this limitation, Compressed Sensing (CS) has recently been proposed as a viable low complexity signal processing solution for different media compression/reconstruction, providing a systematic approach for reconstructing sparse signals from a small number of random linear observations [21]–[24]. The CS based schemes provide: i) low encoding complexity, since the basis in which the signal is sparse does not need to be computed at the encoder, ii) universality, since the sensing is blind to the source distribution, and iii) privacy preservation [25], since it inherently involves a randomization process comparable to the randomization techniques commonly used in data mining for privacy preservation [26]. More importantly, these schemes allow the progressive compression of 3D objects, where an early coarse approximation can subsequently be improved by simply transmitting additional random linear combinations. The authors in [27], exploited the sparse Laplacian eigen-domain structure of the mesh, by employing conventional CS schemes. However, as the number of vertices of the 3D model grows, Laplacian based compression/reconstruction methods become infeasible, since they require the inversion and the eigenvalue decomposition of matrices, with sizes that are equal to the number of mesh vertices.

To address this issue the 3D object should be divided and processed in submeshes, as suggested in [28]. Although, the proposed method requires the knowledge of the Laplacian eigenvectors (graph Fourier basis vectors) on the decoder side only, the previous approaches (e.g., [19], [20]) require this knowledge both at the encoder and the decoder. This actually means that either they require to calculate the eigenvectors at the encoder side and transmit them to the decoder (which requires both processing and transmission power) or they require to evaluate them in both sides. In this paper, motivated by the aforementioned open issues, we introduce a novel geometry compression/reconstruction algorithm that enhances the benefits of Laplacian, by processing dense 3D models in parts and exploiting during reconstruction, specific local characteristics (e.g., potential correlations of the principal spectral values) of the geometric information of the different parts in the graph Fourier domain. The contribution of this paper can be summarized as follows:

1) We propose a fast and efficient CS based 3D model compression approach, that enables aggressive compression ratios, without introducing significant loss on the visual quality. The proposed approach is superior in terms of computational complexity as compared to spectral compression schemes, since it requires two orders of magnitude less operations for providing compact representations of the 3D meshes, while at the same time achieves similar reconstruction quality.
2) For the first time, we propose the application of model based Bayesian learning to the problem of 3D mesh compression. We derive a novel reconstruction scheme that instead of simply exploiting sample sparsity in the eigendomain (e.g., [27]), we exploit the power law decaying structure of the graph Fourier transform (GFT) coefficient by *fitting an exponentially decaying model* to the projected values of the Euclidean coordinates in the GFT domain. This approach allows us to reduce the degrees of freedom of the compressible graph Fourier coefficients, by permitting only certain configurations of the large and small/zero graph Fourier coefficients providing two immediate benefits as compared to classical CS:
    a) The proposed scheme enables us to reduce significantly the number of measurements required to reconstruct a 3D object, without introducing realizable visual errors.
    b) During spectral coefficient recovery, the scheme allows us to better differentiate useful information from recovery artifacts, leading to a more robust recovery.
3) Finally, to improve the performance of the proposed approach in highly under-determined problems, we exploit the local smoothness of the individual submeshes by minimizing the inward normal values. We analytically demonstrate that this approach, reduces even more the random linear combinations required for accurate recovery.

An *extensive performance assessment using a large collection of different 3D models*, including both dense (with more than 10^4 vertices) and very dense (with more than 3×10^5 vertices), clearly shows the benefits of our method as compared to spectral based approaches (e.g., [18]–[20], [27]) in terms of reconstruction accuracy, compression efficiency and complexity. By visualizing the visual error on the reconstructed models, we conclude that the benefits of the proposed schemes become more apparent in very dense meshes, where the local characteristics are better preserved.

B. Organization

The remainder of this paper is organized as follows. In Section II, we briefly review concepts and terminology related to the GFT-based compression. In Section III, we present the proposed encoding architecture together with the developed reconstruction algorithm. In Section IV, the performance of the proposed system is evaluated and compared to state-of-the-art approaches, by taking into account different 3D models. Finally, Section V concludes this paper.

C. Notation

The entry in the i-th row and j-th column of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}_{(i,j)}$, while the i-th row and j-th column is denoted by $\mathbf{A}_{(i,:)}$, $\mathbf{A}_{(:,j)}$ respectively. $(\cdot)^T$ denotes transposition; $\mathbb{E}[\cdot]$ denotes the expectation operator; the trace of a matrix $\mathbf{A}$ is denoted by $Tr(\mathbf{A})$.

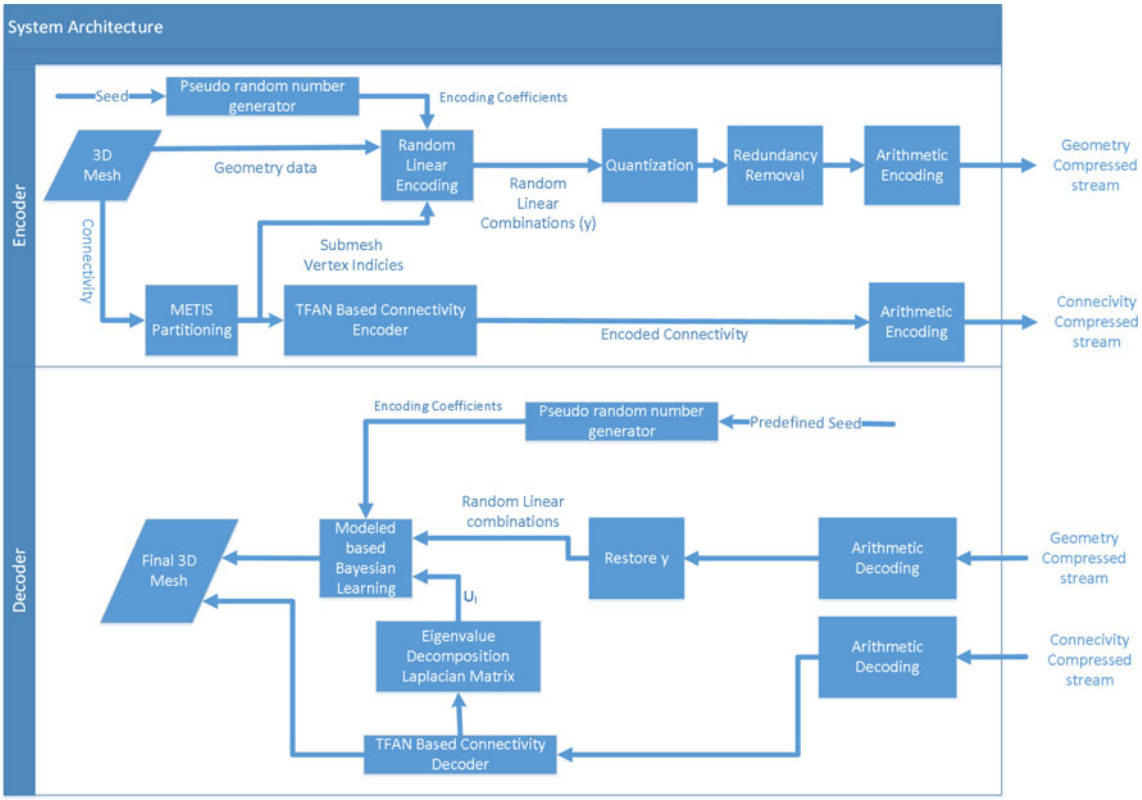


Fig. 1. Encoder and decoder architecture.

II. 3D Mesh Compression Using Graph Fourier Transform

In this work we focus on triangle meshes, which are the most common polygon models. Let us assume that each triangle mesh $\mathcal{M}$ with n vertices can be represented by two different sets $\mathcal{M} = (V, F)$ corresponding to the vertices (V) and the indexed faces (F) of the mesh. A set of edges (E) can be directly derived from V and F. Most mesh geometry compression works, e.g., [18], [19], [29] are based on the fact that smooth geometries should yield spectra, dominated by low frequency components and suggest projecting the Cartesian coordinates $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \Re^{n\times 1}$ in the graph Fourier basis spanned by the eigenvectors $\mathbf{u}_i$ of the Laplacian operator $\mathbf{L}$. This matrix is calculated as follows:

$$\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-1}\mathbf{C} \tag{1}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix, and $\mathbf{C} \in \Re^{n\times n}$ is the connectivity matrix of the mesh with elements

$$\mathbf{C}_{(i,j)} = \begin{cases} 1, & (i,j) \in (E) \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

$\mathbf{D}$ is the diagonal matrix with $\mathbf{D}_{(i,i)} = |N(i)|$ and $N(i) = \{j|(i,j) \in (E)\}$ is a set with the immediate neighbors for node i. Let as assume that the eigenvalue decomposition of $\mathbf{L}$ is written as

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{3}$$

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 $\mathbf{u}_i \in \Re^{n\times 1}$ that span the graph Fourier basis. Then the aforementioned compression schemes take advantage of the fact that the projection of Euclidean coordinates (e.g., $\mathbf{v}$) to the eigenvectors $\mathbf{U}$ of the Laplacian operator $\mathbf{L}$ results in sparse representations $\mathbf{s}_v = \mathbf{U}^T\mathbf{v}$ that allows the following compact representation:

$$\hat{\mathbf{v}} \approx \sum_{i=1}^{k} \left(\mathbf{u}_i^T \mathbf{v}\right) \mathbf{u}_i, \quad k \leq n. \tag{4}$$

III. Proposed Architecture

In this section, we initially present the proposed schemes for compressing/reconstructing dense 3D models. A block diagram of the proposed architecture is presented in Fig. 1. According to the presented architecture, the geometry and the connectivity of the mesh are treated individually. Regarding the connectivity, we have used a state-of-the-art connectivity encoder (e.g., TFAN encoder [17]), which is the one that was also used by many state-of-the-art 3D mesh compression algorithms (e.g., Open3DGC [17]) and we thus focus on providing a novel approach for compressing/reconstructing the geometry of the mesh. The proposed geometry compression algorithm exhibits similar performance to the spectral compression approach (GFT), while minimizing the processing at the encoder side, since it doesn't require the computation of the basis in which the signal is sparse. In addition, it is "universal", which means that it is completely blind to the source distribution. The exploitation of key characteristics of specific 3D models (e.g., structure of the GFT coefficients) is performed at the decoder side. A detailed description of the

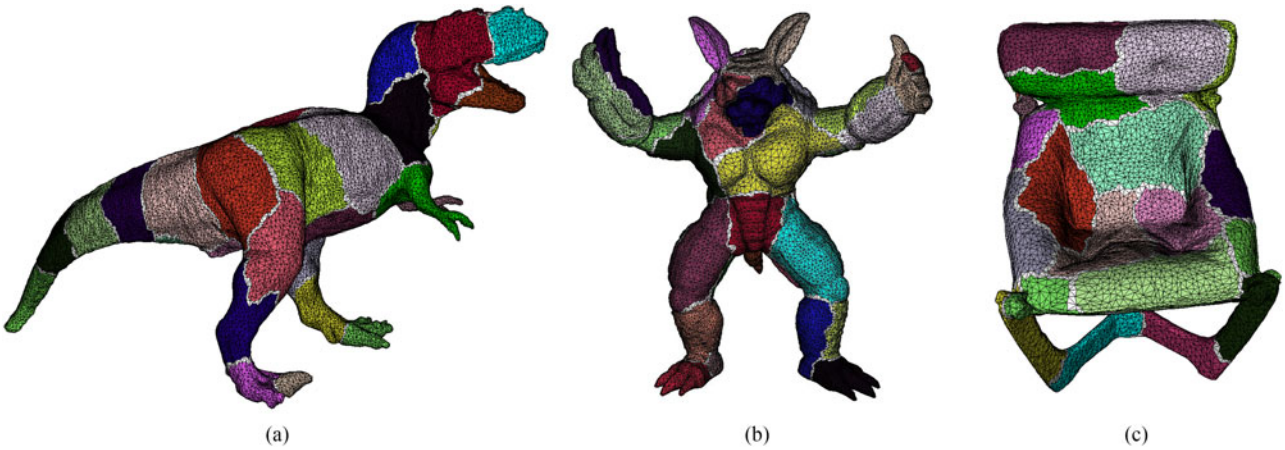


Fig. 2. Graph partitioning using MeTiS. Submeshes are colored with randomly selected colors. Triangles with vertices that belong to different submeshes are colored white. (a) Stanford tyrannosaurus model, 20 002 vertices, 60 000 edges, 30 submeshes. (b) Stanford armadillo model, 20 002 vertices, 60 000 edges, 30 submeshes. (c) A chair 15 340 vertices, 45 990 edges, 30 submeshes.

compression/reconstruction approaches is presented in the section that follows.

A. Partitioning and Processing of Submeshes at the Encoder

At the encoder side, the original 3D mesh is divided into L submeshes by using the MeTiS method described in [30]. MeTis algorithm provides a viable option for solving graph partitioning problems of large-scale graphs (e.g. graphs containing over 100,000 vertices), offering a linear time optimized implementation. MeTiS efficiently addresses the general partitioning objective by computing an L-way partitioning such that the number of edges that straddle different partitions (i.e., the *edge-cut*), is minimized. Each submesh l consist of n_l nodes, where $\sum_{l=1}^{L} n_l = n$. The geometry of the n_l nodes included in the l-th submesh is represented as a matrix of size $n_l \times 3$, $\mathbf{v}_l = [\mathbf{x}_l, \mathbf{y}_l, \mathbf{z}_l]$, where $\mathbf{x}_l, \mathbf{y}_l, \mathbf{z}_l \in \Re^{n_l \times 1}$. Fig. 2 illustrates partitions generated by MeTiS for the Stanford Armadillo, Stanford Tyrannosaurus and a Chair model scanned with a Kinect sensor.[1] Submeshes are colored with randomly selected colors, while the triangles with vertices that belong to different submeshes are colored white.

The trade-off included in not processing the geometry of the 3D object as a single mesh, is the degradation in the reconstruction quality along the submesh endpoints, so called as boundary nodes. This degradation is attributed to the fact that we ignore the neighbors of the boundary nodes that are not included in the submesh that we process. To overcome this problem, we suggest to process the geometry of overlapped submeshes, by extending each submesh with the neighbors of the boundary nodes of adjacent submeshes. To be more specific, if $I_l = \{i_1, \ldots, i_{n_l}\}$ denotes the set of indices of the 3D mesh nodes belonging in submesh l

$$\mathbf{v}_{(I_l,:)} = \left[\mathbf{v}_{I_1}^T, \ldots, \mathbf{v}_{I_{n_l}}^T\right]^T, \quad \mathbf{v}_i = [\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i] \tag{5}$$

and $I_{l_b} = \{i_{l_1}, \ldots, i_{l_b}\} \subset I_l$ is the set of indices of the boundary nodes of submesh l, then we suggest extending matrix $\mathbf{v}_l$ with the Cartesian coordinates of the neighbors of the boundary nodes that belong to the set $I_{\mathrm{nb}} = \cup_{j=1}^{b} N(i_{l_j}) \setminus I_l$

$$\mathbf{v}_{l_e} = \left[\mathbf{v}_{(I_l,:)}^T, \mathbf{v}_{(I_{\mathrm{nb}},:)}^T\right]^T \tag{6}$$

where $|I_l \cup I_{\mathrm{nb}}| = n_{l_e} > n_l$ vertices.

For each submesh, the source generates $M \times 3$ random linear combinations[2] (see Fig. 1) by using a random matrix $\mathbf{A}$ of dimension $M \times n_{l_e}$, that consist of ± 1 values selected with probability 0.5 (i.e, Rademacher distribution)[3] (Random Linear Coding - RLC) as follows:

$$\mathbf{y} = \mathbf{A}\mathbf{v}_{l_e}. \tag{7}$$

The geometry of each submesh is therefore encoded using an $M \times n_{l_e}$ (with $M \leq n_{l_e}$) random linear matrix $\mathbf{A}$ that is constructed both at the encoder and decoder using a pseudo-random number generator (PRNG) that generates a sequence of numbers that approximate the properties of random numbers. The generated sequence is completely determined by a random seed, represented by a single real number that is the only information that has to be transmitted to the receiver side. To reduce even more the communication requirements, the same seed is used for encoding all the submeshes composing the original mesh. Therefore, even if we take into account the overhead for transmitting this seed, the calculated compression ratio at the decoder is not affected at all, since the same seed is used for encoding a total of $n = \sum_{l=1}^{L} n_l > 10^5$ vertices, each represented by three 32-bit numbers corresponding to the x, y, z Euclidean coordinates.

The final encoding stage further removes any remaining redundancy between consecutive vertex coordinates within a submesh through a redundancy removal module, which processes the difference between the encoded coordinates. The output of

[1]The Chair Model was retrieved from the kscan3d gallery. [Online]. Available: http://www.kscan3d.com/gallery/

[2]The value of M determines the achieved compression ratio as it is shown in Section IV.

[3]Note that each column of $\mathbf{v}_{l_e}$ is treated individually.

this module, is subsequently uniformly quantized yielding the n_{l_e} -dimensional vector $\mathbf{z}_q$. The latter is encoded using lossless Huffman coding to produce the encoded vector $\mathbf{c}$.

At the decoder side, after the restoration of the random linear combinations by decoding the differences between coordinates, the vector with the RLC $\mathbf{y}_q$ may be written as

$$\mathbf{y}_q = Q\left(\mathbf{A}\mathbf{v}_{l_e}\right) = \mathbf{A}\mathbf{v}_{l_e} + \mathbf{w}_q \tag{8}$$

$$= \mathbf{A}_{u_l}\mathbf{s}_{l_e} + \mathbf{w}_q \tag{9}$$

where $\mathbf{A}_{u_l} = \mathbf{A}\mathbf{U}_{l_e}$, $\mathbf{U}_{l_e}$ are the eigenvectors of the Laplacian operator $\mathbf{L}_{l_e}$ of the extended submesh l, $\mathbf{s}_{l_e} = [\mathbf{s}_{x_{l_e}}, \mathbf{s}_{y_{l_e}}, \mathbf{s}_{z_{l_e}}] \in \Re^{n_{l_e}\times 3}$ are the projected Cartesian coordinates in the corresponding graph Fourier basis and $\mathbf{w}_q = [\mathbf{w}_{q_x}, \mathbf{w}_{q_y}, \mathbf{w}_{q_z}]$ represents the quantization error. Before proceeding to the reconstruction of points $\mathbf{v}_{l_e}$ from $\mathbf{y}_q$, we construct the Laplacian matrix $\mathbf{L}_{l_e}$ directly from the decoded connectivity and we evaluate matrix $\mathbf{U}_{l_e}$ for each submesh, by performing a singular value decomposition. Then, the decoder generates the binary matrix $\mathbf{A}$ used for encoding the vertices of each submesh, by using the predefined random seed. Both the generated encoding matrix $\mathbf{A}$, the matrix with the Laplacian eigenvectors $\mathbf{U}_{l_e}$ and the vectors with the RLC $\mathbf{y}_q$ are used to decode the original geometry information of each submesh l, by applying the algorithm presented in the section that follows.

B. Reconstruction via Model-Based Bayesian Learning

Motivated by the fact that: i) the behavior of the GFT is very similar to the DCT since it redistributes the energy contained in the data, so that most of energy is contained in a small number of components [19] and ii) the DCT coefficient values of natural images and audio signal are usually modeled as multivariate Gaussian distributions [31], we assume that the projection of the Cartesian coordinates of each extended submesh l in the GFT domain (e.g., $\mathbf{s}_{l_e} = \mathbf{U}_{l_e}^T\mathbf{v}_{l_e}$) can be well approximated by a sparse vector with k non-zero components and $n_{l_e} - k$ zeros

$$\mathbf{s}_{i_{l_e}} = \left[\mathbf{s}_{i_{l_k}}, \mathbf{0}_{n_{l_e}-k}\right]^T, \;\; \mathbf{s}_{i_{l_k}} = \left[s_{i_{l_1}}, \ldots, s_{i_{l_k}}\right], k < n_{l_e} \tag{10}$$

where i corresponds to the x, y, or z coordinates and $\mathbf{s}_{i_{l_k}}$ denotes the non-zero block of size k that can be modeled as a parametrized multivariate Gaussian distribution

$$p\left(\mathbf{s}_{i_{l_k}}\right) \sim \mathcal{N}\left(0, \mathbf{C}_{i_0}\right), \;\; \mathbf{C}_{i_0} = \gamma_i\Sigma_i, \; i = \{x, y, z\} \tag{11}$$

where γ_i is a scalar parameter and $\Sigma_i \in \Re^{k\times k}$ is a positive definite matrix.

By using the Bayes rule and assuming that the noise vector $\mathbf{w}_{q_i}$ in (9) can be considered as an $M \times 1$ vector with Gaussian i.i.d. random variables $\mathbf{w}_{q_i} \sim \mathcal{N}(0, \sigma_{w_i}\mathbf{I}_M)$ we obtain the following proposition for evaluating the posterior probability given the encoding coefficients:

Proposition 1: The posterior density of $\mathbf{s}_{i_{l_k}}$, is also Gaussian $p(\mathbf{s}_{i_{l_k}}|\mathbf{y}_{q_i}; \sigma_{w_i}, \gamma_i\Sigma_i) \sim \mathcal{N}(\mu_{s_{i_l}}, \mathbf{C}_i)$ with the following mean and covariance matrix:

$$\mu_{s_{i_l}} = \mathbf{C}_{i_0}\mathbf{A}_{u_{l_k}}^T\left(\mathbf{A}_{u_{l_k}}\mathbf{C}_{i_0}\mathbf{A}_{u_{l_k}}^T + \sigma_{w_i}\mathbf{I}_M\right)^{-1}\mathbf{y}_{q_i} \tag{12}$$

$$\mathbf{C}_i = \mathbf{C}_{i_0} - \mathbf{C}_{i_0}\mathbf{A}_{u_{l_k}}^T\left(\mathbf{A}_{u_{l_k}}\mathbf{C}_{i_0}\mathbf{A}_{u_{l_k}}^T + \sigma_{w_i}\mathbf{I}_M\right)^{-1}\mathbf{A}_{u_{l_k}}\mathbf{C}_{i_0} \tag{13}$$

where $\mathbf{A}_{u_{l_k}} = \mathbf{A}_{u_{l(:,1:k)}}$ is an $M \times k$ matrix that consists of the first k columns of $\mathbf{A}_{u_l}$.

Proof: The proof is given at the Appendix A. ■

Thus, given the parameters $\sigma_{w_i}, \gamma_i, \Sigma_i, i = \{x, y, z\}$ the maximum a posteriori (MAP) estimate of the Cartesian coordinates of the extended submesh l is given by

$$\hat{\mathbf{v}}_{l_e} = \mathbf{U}_{l_e}\begin{bmatrix}\mu_{s_{x_l}} & \mu_{s_{y_l}} & \mu_{s_{z_l}} \\ \mathbf{0}_{n_{l_e}-k} & \mathbf{0}_{n_{l_e}-k} & \mathbf{0}_{n_{l_e}-k}\end{bmatrix}. \tag{14}$$

To find the parameters $\sigma_{w_i}, \gamma_i, \Sigma_i$, we employ the expectation-maximization (EM) algorithm to maximize $p(\mathbf{y}_{q_i}; \sigma_{w_i}, \gamma_i, \Sigma_i)$ per coordinate, meaning that i can be x, y or z. This is equivalent to minimizing $-\log(p(\mathbf{y}_{q_i}; \theta))$, yielding the following marginal likelihood of the observed data:

$$\mathcal{L}\left(\theta; \mathbf{y}_{q_i}\right) = \left(\mathbf{y}_{q_i}\right)^T\Sigma_{y_i}^{-1}\mathbf{y}_{q_i} + \log|\Sigma_{y_i}| \tag{15}$$

where $\Sigma_{y_i} = \sigma_w\mathbf{I}_M + \mathbf{A}_{u_{l_k}}\mathbf{C}_{i_0}\mathbf{A}_{u_{l_k}}^T$. The following proposition provides an iterative scheme for estimating the model parameters $\sigma_{w_i}, \gamma_i, \Sigma_i$ that maximize the aforementioned likelihood

Proposition 2: The application of the EM algorithm per coordinate $i = \{x, y, z\}$ leads to the following learning rules for the model parameters $\sigma_{w_i}, \gamma_i, \Sigma_i$:

$$\sigma_{w_i} = \frac{\left\|\mathbf{y}_{q_i} - \mathbf{A}_{u_{l_k}}\mathbf{s}_{i_{l_k}}\right\|_2^2 + \sigma_{w_i}\left[k - Tr\left(\mathbf{C}_i\mathbf{C}_{i_0}^{-1}\right)\right]}{M} \tag{16}$$

$$\gamma_i = \frac{Tr\left(\Sigma_i^{-1}\left(\mathbf{C}_i + \mu_{s_{i_l}}\left(\mu_{s_{i_l}}\right)^T\right)\right)}{k} \tag{17}$$

$$\Sigma_i = \frac{\mathbf{C}_i + \mu_{s_{i_l}}\left(\mu_{s_{i_l}}\right)^T}{\gamma_i}. \tag{18}$$

Proof: The proof is given in Appendix B. ■

The performance of the proposed algorithm can be further improved by constraining the matrix Σ_i to have a Toeplitz symmetric structure with elements $(\Sigma_i)_{(m,l)} = r_i^{|m-l|}, \forall m, l \in [1, \ldots, k]$. This form is equivalent to modeling the elements in the non-zero block as a first order auto-regressive process. The value of r_i can be estimated by

$$r_i = \text{sign}\left(\text{m}_1/\text{m}_0\right)\min\left\{\left|\text{m}_1/\text{m}_0\right|, 0.99\right\} \tag{19}$$

where m_0 is the average of the elements along the main diagonal and m_1 is the average of elements along the main sub-diagonal of Σ_i, 0.99 is a bound selected by the user. The proposed geometry reconstruction algorithm is summarized in Table I.

TABLE I
MESH RECONSTRUCTION VIA MODEL BAYESIAN LEARNING (MBL)

MBL Recovery:

Inputs: Encoding Matrices and GFT vectors : $\mathbf{A}$, $\mathbf{U}_{l_e}$
Encoded Samples: $\mathbf{y}_q$, Non zero Block length k,
Output: Estimated Cartesian Coordinates $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \Re^{n_{l_e} \times 1}$

For each Submesh $l = 1, \ldots, L$**,**
  For each iteration $m = 1, \ldots, K$
    a. Evaluate non zero values $\mu_{s_{x_l}}, \mu_{s_{y_l}}, \mu_{s_{z_l}}$ from (12)
    b. Evaluate the corresponding variances $\mathbf{C}_x, \mathbf{C}_y, \mathbf{C}_z$ via (13)
    c. Update $\sigma_{w_i}, \gamma_i, \Sigma_i$ for every coordinate i=x,y,z from (16)-(18)
    d. Update the value of r_i from (19) and re-evaluate
      $\Sigma_i = Toeplitz\{[1\ r\ \ldots r^{k-1}]\}$.
  end For
  Evaluate Cartesian Coordinates $\hat{\mathbf{v}}_{l_e}$ of the l submesh from (14)
  and drop the points that belong to the set $\cup_{j=1}^{b} N(i_{l_j}) \setminus I_l$
  where i_{l_j} denotes the indices of the boundary nodes in submesh l.
end For

TABLE II
CONSIDERED ALGORITHMS/ACRONYMS

Algorithms	Acronyms	Encoded Samples (M)
Spectral Compression [19], [28]	GFT	Number of Spectral Components
Least Square Meshes [18]	LSM	Number of Anchor Points
GFT based Compressed Sensing based on [27]*	CS GFT	Number of Random Linear Combinations
Model based Bayesian Learning	MBL	Number of Random Linear Combinations

* A modified version of [27] have been implemented, that in contrast to [27] process the submeshes in parts and that exploits the local smoothness as described in Section III-C.

C. Exploiting the Local Smoothness

The compression efficiency of the aforementioned approaches can be improved by allowing reconstruction of the k principal GFT coefficients from a smaller number of linear combinations M. However, as M decreases, the likelihood function in (15) has singularities, which are attributed to the rank reduction of the Gram of the encoding matrix $\mathbf{A}_{u_{l_k}}$. This effect can be avoided by adding one more constraint at the vertices of each submesh to distribute them fairly in each submesh. This can be achieved by moving its vertex to its center of gravity [18]. This constraint can be expressed by the following equation for vertex i:

$$\mathbf{v}_i - \frac{1}{|N(i)|} \sum_{j:(i,j)\in(E)} \mathbf{v}_j = 0. \tag{20}$$

Provided that the connectivity information is assumed to be known at the receiver, the aforementioned constraint can be satisfied by solving the following linear systems for each submesh l

$$\mathbf{L}_{l_e} \mathbf{U}_{l_e} \mathbf{s}_{i_{l_e}} = \mathbf{0}_{n_{l_e} \times 1},\ i = x, y, z. \tag{21}$$

Therefore, we suggest executing the algorithm of Table I by using the following extended versions of the encoded vector and its corresponding encoding matrix:

$$\mathbf{y}'_{q_i} = \begin{pmatrix} \mathbf{y}_{q_i} \\ \mathbf{0}_{n_{l_e} \times 1} \end{pmatrix},\ \ \mathbf{A}'_{u_{l_k}} = \begin{pmatrix} \mathbf{A}_{u_{l_k}} \\ \mathbf{L}_{u_{l_k}} \end{pmatrix} \tag{22}$$

where $\mathbf{L}_{u_{l_k}}$ is an $n_{l_e} \times k$ matrix that is formed by the first k columns of matrix $\mathbf{L}_{l_e} \mathbf{U}_{l_e}$. This extension allows an accurate reconstruction from a reduced number of random linear combination, since it improves the condition number of the matrix $\mathbf{A}_{u_{l_k}} \mathbf{C}_{i_0} \mathbf{A}^T_{u_{l_k}}$ that determines the MAP estimation accuracy. The benefits of adding this constraint are evaluated in the section that follows, while the following proposition provide a theoretical proof that matrix $\mathbf{A}'_{u_{l_k}}$ will satisfy the requirements for ensuring an accurate reconstruction at the decoder.

Proposition 3: The column-normalized matrix $\tilde{\mathbf{A}} = \frac{1}{\sqrt{M}} \mathbf{A}'_{u_{l_k}} \in \Re^{(M+n_{l_e}) \times k}$ satisfies the restricted isometry property (RIP) with a constant $\rho_s = \rho_k + \lambda^2_{\max}/M \in (0, 1)$, where ρ_k is the constant of the RIP for the matrix $\mathbf{A}_{u_{l_k}}$ and $\lambda_{\max}$ is the maximum eigenvalue of the Laplacian matrix $\mathbf{L}_{u_{l_k}}$.

Proof: The proof is provided in Appendix C. ■

IV. SIMULATION RESULTS

The focus of this study is: 1) to evaluate the benefits of processing overlapped submeshes as compared to the non-overlapping case 2) to identify the benefits of the proposed compression/reconstruction schemes as compared to the traditional compression approaches. The proposed schemes are studied by using a large variety of different 3D objects such as the Stanford models and models reconstructed from range scans.

A. Experimental Setup and Metrics

We assume that each object is divided into L submeshes, where L is a parameter selected by the user. Then, each submesh is compressed either by using the conventional or the CS schemes. As conventional schemes we consider: 1) the approaches that provide compact representation of 3D meshes in the GFT basis (e.g., [19]), presented in Section II, 2) the approach of Sorkine and Cohen-Or [18] known as least square meshes (LSM), that make use of specific points in the mesh so called as anchor points. The CS schemes include: 1) the method that was proposed in [27] that reconstructs each submesh from RLCs, by exploiting the sample sparsity of each submesh in the GFT domain (CS GFT) 2) the proposed method presented in Table I. At this point it should be mentioned that the number of the selected anchor points in the LSM case, was equal to the number of selected random linear combinations in the CS approaches. To allow a more accurate reconstruction the anchors were selected randomly from the set of the boundary nodes of each submesh.

The aforementioned methods are evaluated in terms of both compression efficiency and reconstruction accuracy. The compression efficiency of the proposed schemes is evaluated by using the Compression Ratio (CR)

$$CR = 1 - \frac{\sum_{l=1}^{L} M_l \times q}{n \times 32} \tag{23}$$

where M_l are the low frequency components (GFT Case), the generated encoded samples for submesh l (CS GFT and MBL cases), or the number of anchor points (LSM case) and q is the

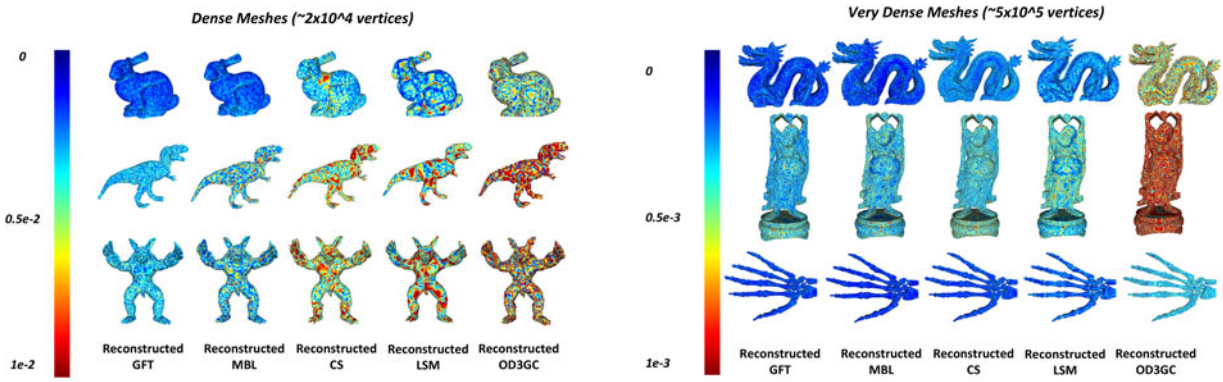


Fig. 3. Visualization of the visual error in the reconstructed 3D objects. The compression efficiency is equal to 2.3bpv.

number of bits used for the representation of the encoded samples. The quantization is performed by applying the Lloyd max algorithm [32]. The reconstruction effectiveness is evaluated by the normalized mean square visual error (NMSVE) defined in [19], as the average error in the Cartesian and Laplacian domains

$$NMSVE = \frac{1}{2n}\sum_{i=1}^{n}\left(\|\mathbf{v}_i - \tilde{\mathbf{v}}_i\|_2 + \|GL(\mathbf{v}_i) - GL(\tilde{\mathbf{v}}_i)\|_2\right) \tag{24}$$

where d_{ij} denotes the Euclidean distance between i and j and $GL(\mathbf{v}_i) = v_i - \sum_{j\in N(i)} d_{ij}^{-1} v_j / \sum_{j\in N(i)} d_{ij}^{-1}$.

B. Performance Evaluation

Evaluation of MBL versus state-of-the-art approaches presented in Table II: In Fig. 3 we visualize the visual error on dense ($>2\times10^4$ vertices) and very dense meshes ($>3\times10^5$ vertices) that have been reconstructed using the approaches presented in the original versions as well as the Open3DGC method of [17]. By inspecting this figure, it is obvious that the MBL method can achieve CRs that correspond to even 2.3 bit per vertex (bpv) with extremely low encoding complexity (by performing only additions) and without affecting significantly the visual error of the reconstructed model. By inspecting also the error scale, it can be easily concluded that the benefits become more apparent in very dense meshes since the maximum error values, as compared to those using dense meshes, is lower by one order of magnitude. Finally, the OD3GC method that performs direct quantization to the 3-space coordinates, introduces high-frequency errors into the model that modify significantly the appearance of the surface, resulting in a model with a blocky structure, where the reconstruction errors are highly noticeable. Therefore, in the rest part of this sections, we focus only on comparing our results with relevant approaches (e.g., the approaches of Table II) that introduce low frequency errors to the model, enabling aggressive compression ratios without introducing significant loss on the visual quality.

Segmentation effects: To evaluate the benefits of processing overlapping submeshes, as compared to the non-overlapping case presented in [19], we executed the GFT method for the same number of low frequency components per submesh. In Fig. 4 we provide the reconstructed Tyrannosaurus models, where it is clearly shown that the overlapping method described in Section II results in a more accurate reconstruction. This is due to the fact that the boundary points (vertices of red triangles) are recovered almost perfectly.

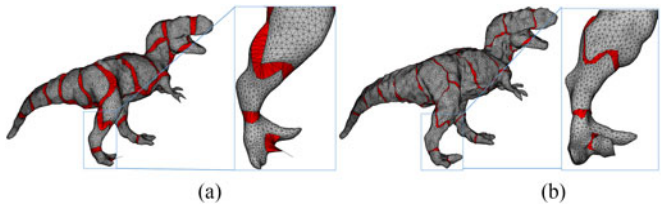


Fig. 4. Reconstruction of the mesh geometry using (a) nonoverlapped (NMSVE: −27 dB) and (b) overlapped submeshes (NMVSE: −34 dB). The achieved CR in both cases is 0.92.

Evaluation of reconstructed visual quality for different CRs: In Fig. 5(a), the obtained NMSVE for a chair model scanned by using a Kinect sensor, is plotted against the achieved CR after transmitting the $M = [65, 130, 195, 250, 280]$ lower frequency components per submesh, in the GFT case, M random linear combinations in the CS cases, or M anchor points in the LSM case. The number of bits for representing transmitted samples were selected equal to $q = 12$. By inspecting the figure, it is clear that the application of the proposed MBL algorithm at the decoder reduces the number of the transmitted samples M required for the efficient reconstruction of the 3D mesh, with respect to the CS GFT and LSM approaches, while achieving performance almost similar to the conventional one (GFT). The main difference between the two approaches (GFT and MBL) is that the GFT allows the reconstruction of the largest k spectral components which in some cases are not the initial k (low frequency components). In the MBL approach on the other hand, we reconstruct accurately the k-lower components and the rest are set to zero. Therefore, in cases we process very dense meshes in parts that correspond to quite smooth surfaces the k- largest spectral components coincide with the k - lower spectral components, and thus the performance of GFT and MBL almost coincide. This is also shown in Fig. 6, where we have partitioned the Dragon Model (437.645 vertices) into 1000 parts, is obvious that the performance of MBL is almost similar with that of GFT.

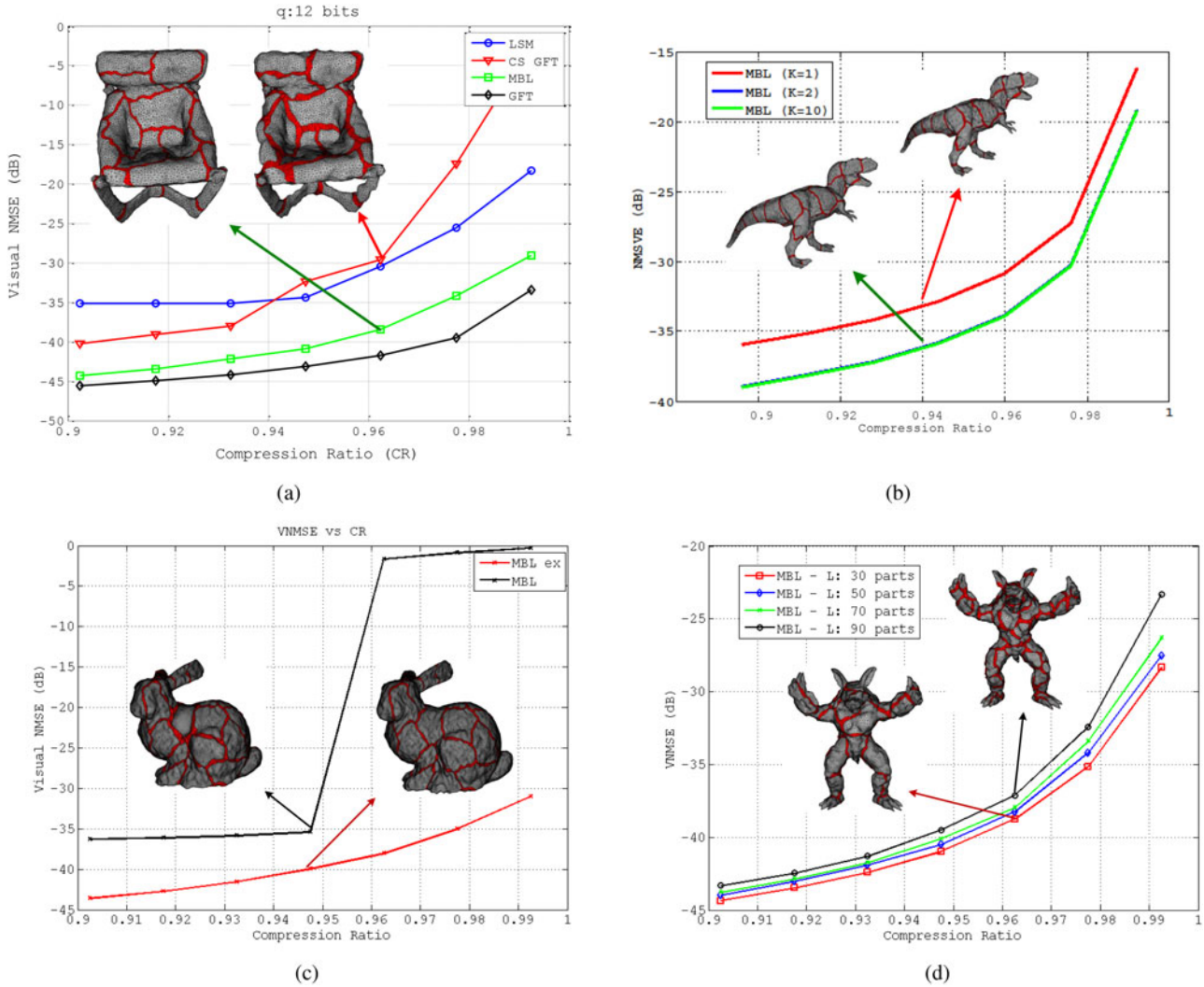


Fig. 5. (a) VNMSE versus CR using the chair model. (b) VNMSE versus CR for different number of iterations. (c) VNMSE versus CR with and without exploiting local smoothness. (d)VNMSE versus CR for different number of submeshes (L).

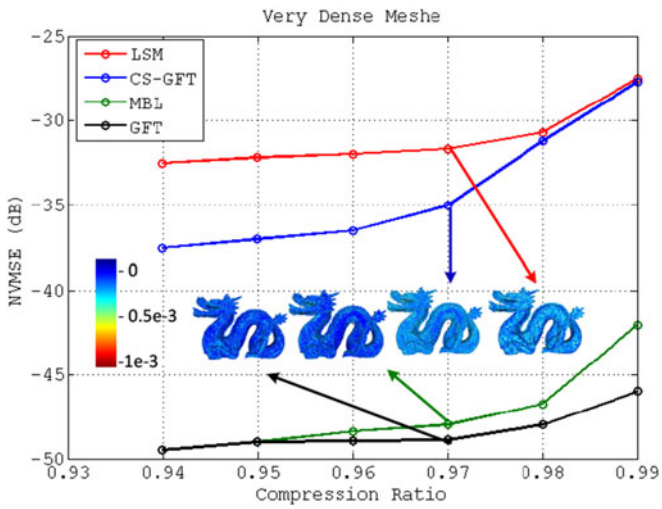


Fig. 6. VNMSE versus CR using the Dragon model (437645 vertices) partitioned in 1000 parts.

At this point, it should be noted that the application of RLC at the CS encoder requires only additions, instead of computing the projection of the coordinates to the Laplacian eigenvectors and selecting the M largest spectral coefficients. In other words, the CS approach requires only $n_{l_e} M - M$ additions and 0 multiplications for compressing the 3D object, while the GFT requires $\mathcal{O}(n_{l_e}^3)$ multiplications and $\mathcal{O}(n_{l_e}^3)$ additions. More importantly, the proposed method inherently involves a randomization process offering privacy preservation without any additional cost [26].

Required iterations for the MBL approach: Fig. 5(b), shows the NMSVE against the number of executed iterations for the Tyrannosaurus Model, where it is clearly shown that the MBL algorithm converges after two iterations.

Benefits of exploiting local smoothness: To evaluate the effects of exploiting the local smoothness of the different submeshes, we executed the MBL algorithm of Table I either by using the encoding vector $\mathbf{y}_q$ and the corresponding encoding

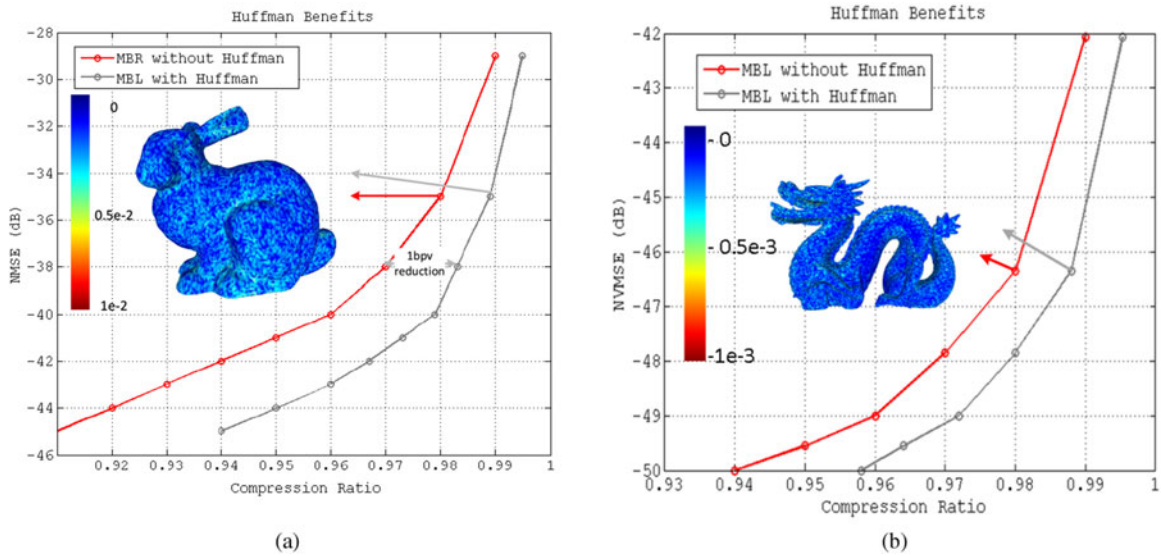


Fig. 7. Benefits of adopting a redundancy removal unit and an entropy encoding unit on the top of our scheme: (a) dense mesh (Bunny Model, 34.817 vertices), and (b) very dense mesh (Dragon Model 437.645 vertices.)

matrix $\mathbf{A}_{u_l}$ per submesh l or by employing their extended versions defined in (22) for different compression ratios (MBL). The evaluated VNMSE's are presented in Fig. 5(c), where it is clearly shown that by distributing the vertices fairly in each submesh we are able to increase significantly the reconstruction accuracy, especially for very high CR values (e.g, $CR > 0.94$).

Fig. 5(d) shows the obtained VNMSE for the Armadillo model against the CR, after partitioning the 3D mesh to a different number of submeshes (e.g., $L = 30, 50, 70, 90$). It is clearly shown that as the number of submeshes increases the VNMSE also slightly increases. This small performance degradation is attributed to the fact that the errors on the boundary nodes are slightly larger as compared to the errors in the internal nodes of each submesh. However, it should be noted that processing small parts reduces significantly the computational complexity of the reconstruction schemes, that is of the order of $O(Ln_{l_e}^3)$, where L corresponds to the number of submeshes and n_{l_e} to the size of the extended submeshes. Therefore, the adopted mesh segmentation method provide a low-cost flexible way to trade off efficiency for performance.

Benefits of using a redundancy removal and an entropy encoding unit: To evaluate the benefits of the redundancy removal and the entropy encoding unit, we provide results with and without the use of the aforementioned blocks. In Fig. 7(a)–7(d) we provide the NMSVE at the output of the quantizer and at the output of the entropy encoder. Note that the proposed schemes with two additional encoding units at the transmitter achieved to improve the compression efficiency by almost 25% without affecting at all the reconstruction quality. The results that we get are similar in both cases, i.e., dense and very dense meshes.

Quantization effects: To study the quantization noise effects on the performance of the presented algorithms, we conducted experiments, assuming different quantization rates. In Fig. 8, we present the obtained NMSVE for the person model (partitioned in 30 submeshes), against the transmitted M lower frequency components per submesh, in the GFT case, M random linear

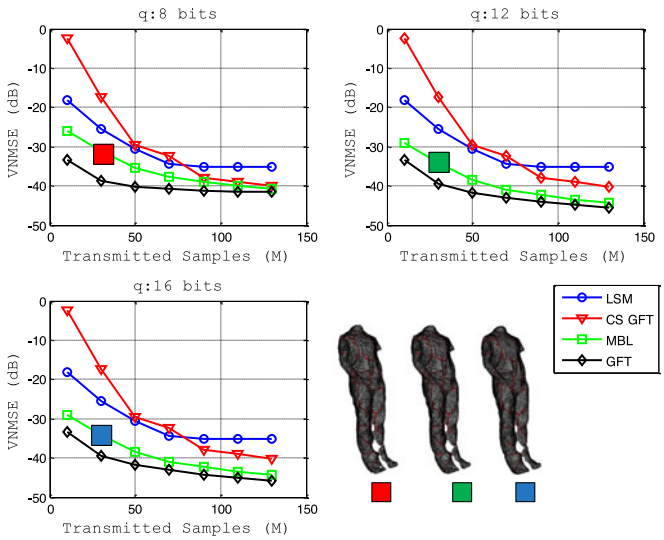


Fig. 8. Quantization noise effects: NMSVE for the person model (partitioned in 30 submeshes), against the transmitted M samples using q = 8, 12, and 16 bits/sample, using a person 3D model reconstructed from range scans.

combinations in the CS cases, or M anchor points in the LSM case, assuming that the quantization bits are equal to $q = 8, 12$ and 16 bits respectively. These figures clearly show that the GFT and the MBL algorithms are more robust to the quantization noise as compared to the LSM and CS GFT algorithm. The LSM algorithm approximates each submesh with smoothed surfaces that minimize the least square error between the noisy anchor points and the original points. As a result, the increased noise strongly affects the error on the boundary points resulting in very low reconstruction quality. In the MBL case, we reduce the degrees of freedom of the sparse/compressible representation of the partitioned surfaces in the corresponding GFT domain, by permitting only certain configurations of the large and zero/small spectral coefficients, providing two immediate benefits as compared to CS GFT. First, we reduce significantly

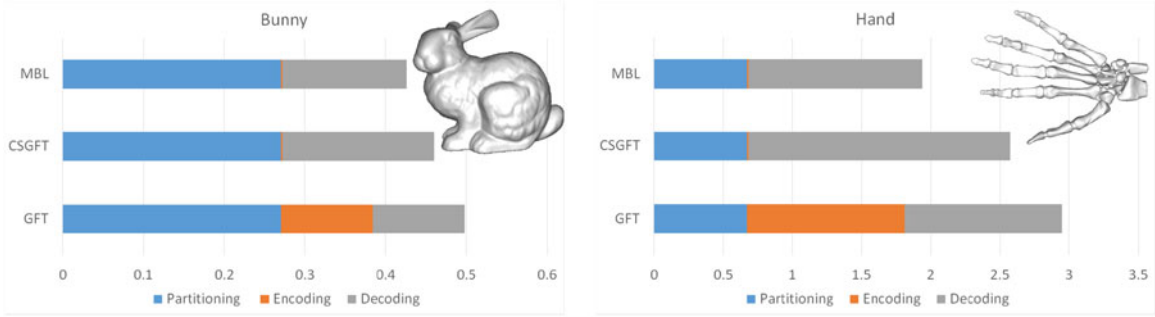


Fig. 9. Execution times (sec) of MBL, CSGFT, and GFT techniques for (a) the Bunny model (v34.817) and (b) the Hand model (v327.323). Blue is the MeTiS partitioning time for 272 and 2557 partitions, respectively. Orange is the compression execution time, which is extremely small in MBL and CSGFT techniques compared to the decompression time, shown with gray color.

TABLE III
PROPERTIES OF THE CONSIDERED ALGORITHMS

	Algorithms			
Properties	GFT	LSM	CS GFT	MBL
Low Encoding Complexity	✗	√	√	√
Compression Universality	✗	√	√	√
Privacy Preservation	✗	✗	√	√
Compression Efficiency	High	Low	Low	High

TABLE IV
COMPLEXITY COMPARISON FOR MESH COMPRESSION TECHNIQUES (IN FLOPS)

Method	Encoder	Decoder
MBL (Proposed algorithm)	$n_{l_e} M - M$	$6k^3 + k^2(4M+3) + k(2M^2 + 3M + 2n_{l_e} - 1) + M^3 + 2M^2 + 2M - n_{l_e} - 1 + \mathcal{O}(n_{l_e}^3)$
Least-square Meshes (LSM) [18]	-	$\frac{7n_{l_e}^3}{3} + 5n_{l_e}^2 - \frac{13n_{l_e}}{3} + 1$
CS GFT [27]	$2n_{l_e} M - M$	$\mathcal{O}(n_{l_e}^3)$
GFT [19] & [28]	$\mathcal{O}(n_{l_e}^3)$	$\mathcal{O}(n_{l_e}^3)$

the number of measurements M required to stably recover a submesh and second, we better differentiate true spectral information from recovery artifacts, which leads to a more robust recovery. The key ingredient is that we provide a more realistic signal model that goes beyond simple sparsity by codifying the inter-dependency structure among the spectral coefficients.

C. Computational Complexity Analysis

To better demonstrate the encoding efficiency of our method, as compared to the spectral compression approach, we initially study the theoretical complexities of the proposed schemes and then focus on providing timing measurements. In Table IV we provide the computational complexities of the presented schemes in terms of floating-point operations (FLOPs). We consider the encoder and the decoder complexities separately, while in the cases of [19], [28] and the decoder of [27] we provide the computational complexity order, since no specific algorithm is provided in these works.

In order to be fair with our comparisons, we have created a modified version of [27], since the original technique cannot be practically employed for dense meshes. In [19] and [28], the computation of the eigenvectors of the Laplacian matrix is required at the encoder as well as at the decoder, which requires $\mathcal{O}(n_{l_e}^3)$ complexity order, at both sides. Note that the computation of the eigenvectors can be avoided at the decoder, at the expense of communication overhead (i.e. the encoder must transmit at the decoder the already computed eigenvectors), which is not a preferable solution in the case of mobile applications.

Considering the proposed technique, the encoder does only one matrix-vector multiplication (the pseudo-random matrix with the unknown vector). Given that the pseudo-random matrix is composed by ± 1 values, the matrix-vector multiplication requires only additions, i.e. $n_{l_e} M - M$ FLOPs. It is important to mention that the enhanced performance of the proposed technique comes without any significant additional complexity cost at the decoder. More importantly, the complexity of the encoder has been kept linear over the number of unknowns n_{l_e}, favoring the employment of the proposed technique in applications where the encoder is a mobile device with limited computational power and energy constraints.

In addition to the complexity analysis we present a timing analysis of the proposed MBL, CSGFT and the GFT approaches, using two representative meshes: 1) the bunny model consisting of 34.817 vertices (dense case) and 2) the hand model consisting of 327.323 verticed (very dense case). The measurements were conducted on an Intel Core i7-4770 (3.40 GHz) processor with 8GB RAM. The aforementioned techniques were implemented in C++ using the Math Kernel Library (MKL),[4] a well optimized library that speeds up all the operations used by those techniques. Moreover, it should be noted that we executed a parallel processing of the different submeshes, both at the encoder and decoder side using four threads. The histograms with the execution times for partitioning, encoding and decoding, the dense and the very dense meshes, are shown in Fig. 9. The MBLs' total execution time as compared to the GFT is about

[4]Intel, "Math Kernel Library (MKL)," 2016. [Online]. Available: http://software.intel.com/en-us/intel-mkl/

25% faster, while both techniques achieve similar VNMSE errors. The encoding time of the MBL and CSGFT is three orders of magnitude less than GFTs' encoding time and thus it can be hardly distinguished in the MBL and CSGFT bars. In addition, the total execution time of the MBL is lower than CSGFT, while at the same time MBL results in significantly lower VNMSE.

V. Conclusion

In this paper, we presented a novel Bayesian learning based 3D mesh geometry reconstruction algorithm that minimizes the random linear coded samples that are required for transmission so that an accurate 3D reconstruction can be obtained at the receiver, by exploiting key characteristics of the Euclidean coordinates in the GFT domain. The advantages of the proposed schemes as compared to the conventional approaches, is that they achieve competitive CRs, while minimizing the compression complexity using a universal and privacy preserving approach. This property is considered critical for emerging 3D model acquisition schemes on off-the-shelf mobile devices.

Appendix

A. Proof of Proposition 1

Proof: Assuming the linear model of (9) the probability density of the observations given $\mathbf{A}_{u_{l_k}}, \mathbf{s}_{i_{l_k}}$ can be factored due to the noise independence assumption to give

$$p\left(\mathbf{y}_{q_i} \mid \mathbf{A}_{u_{l_k}}, \mathbf{s}_{i_{l_k}}\right) = \prod_{n=1}^{M} p\left(y_{q_i n} \mid (\mathbf{A}_{u_{l_k}})_{(:,n)}, \mathbf{s}_{i_{l_k}}\right)$$
$$= \frac{\exp\left(-\frac{1}{2\sigma_{w_i}}\left\|\mathbf{y}_{q_i} - \mathbf{A}_{u_{l_k}}\mathbf{s}_{i_{l_k}}\right\|_2^2\right)}{(2\pi\sigma_{w_i})^{m/2}} \sim \mathcal{N}\left(\mathbf{A}_{u_{l_k}}\mathbf{s}_{i_{l_k}}, \sigma_{w_i}\mathbf{I}_M\right). \tag{25}$$

The posterior distribution can be computed based on Bayes' rule as

$$p\left(\mathbf{s}_{i_{l_k}} \mid \mathbf{y}_{q_i}, \mathbf{A}_{u_{l_k}}\right) = \frac{p\left(\mathbf{y}_{q_i} \mid \mathbf{A}_{u_{l_k}}, \mathbf{s}_{i_{l_k}}\right) p\left(\mathbf{s}_{i_{l_k}}\right)}{p\left(\mathbf{y}_{q_i} \mid \mathbf{A}_{u_{l_k}}\right)} \tag{26}$$

where the normalizing constant, also known as the marginal likelihood, is independent of the graph Fourier coefficients and is given by $p(\mathbf{y}_{q_i}|\mathbf{A}_{u_{l_k}}) = \int p(\mathbf{y}_{q_i}|\mathbf{A}_{u_{l_k}}, \mathbf{s}_{i_{l_k}}) p(\mathbf{s}_{i_{l_k}}) d\mathbf{s}_{i_{l_k}}$.

By writing only the terms of the posterior in (26) that depends on $\mathbf{s}_{i_{l_k}}$ and "completing the square" we obtain

$$p\left(\mathbf{s}_{i_{l_k}} \mid \mathbf{y}_{q_i}, \mathbf{A}_{u_{l_k}}\right) \propto \exp$$
$$\times\left(-\frac{\left(\mathbf{s}_{i_{l_k}} - \mu_{s_{i_l}}\right)^T\left(\sigma_{w_i}^{-1}\mathbf{A}_{u_{l_k}}^T\mathbf{A}_{u_{l_k}} + \mathbf{C}_{i_0}^{-1}\right)\left(\mathbf{s}_{i_{l_k}} - \mu_{s_{i_l}}\right)}{2}\right) \tag{27}$$

where

$$\mu_{s_{i_l}} = \sigma_{w_i}^{-1}\left(\sigma_{w_i}^{-1}\mathbf{A}_{u_{l_k}}^T\mathbf{A}_{u_{l_k}} + \mathbf{C}_{i_0}^{-1}\right)^{-1}\mathbf{A}_{u_{l_k}}\mathbf{y}_{q_i}. \tag{28}$$

Thus, we recognize the form of the posterior distribution as Gaussian, i.e. $p(\mathbf{s}_{i_{l_k}}|\mathbf{y}_{q_i}, \mathbf{A}_{u_{l_k}}) \sim \mathcal{N}(\frac{1}{\sigma_{w_i}}\mathbf{C}_i\mathbf{A}_{u_{l_k}}\mathbf{y}_{q_i}, \mathbf{C}_i)$, where

$$\mathbf{C}_i = \left(\sigma_{w_i}^{-1}\mathbf{A}_{u_{l_k}}^T\mathbf{A}_{u_{l_k}} + \mathbf{C}_{i_0}^{-1}\right)^{-1} \tag{29}$$
$$= \mathbf{C}_{i_0} - \mathbf{C}_{i_0}\mathbf{A}_{u_{l_k}}^T\left(\sigma_{w_i}\mathbf{I}_M + \mathbf{A}_{u_{l_k}}\mathbf{C}_{i_0}\mathbf{A}_{u_{l_k}}^T\right)^{-1}\mathbf{A}_{u_{l_k}}\mathbf{C}_{i_0}. \tag{30}$$

So given the parameters σ_{w_i}, $\mathbf{C}_{i_0}$ the MAP estimate is the mean of the posterior distribution $p(\mathbf{s}_{i_{l_k}}|\mathbf{y}_{q_i}, \mathbf{A}_{u_{l_k}})$ (28) may be written as

$$\hat{\mathbf{s}}_{i_{l_k}} = \left(\sigma_{w_i}\mathbf{C}_{i_0}^{-1} + \mathbf{A}_{u_{l_k}}^T\mathbf{A}_{u_{l_k}}\right)^{-1}\mathbf{A}_{u_{l_k}}^T\mathbf{y}_{q_i}$$
$$= \mathbf{C}_{i_0}\mathbf{A}_{u_{l_k}}^T\left(\sigma_{w_i}\mathbf{I}_M + \mathbf{A}_{u_{l_k}}\mathbf{C}_{i_0}\mathbf{A}_{u_{l_k}}^T\right)^{-1}\mathbf{y}_{q_i} \tag{31}$$

where the last equation follows the matrix identity $(\mathbf{I} + \mathbf{AB})^{-1}\mathbf{A} \equiv \mathbf{A}(\mathbf{I} + \mathbf{BA})^{-1}$. ■

B. Proof of Proposition 2

Proof: The EM algorithm seeks to find the maximum likelihood estimate of the marginal likelihood by iteratively applying the following two steps (E & M step):

E: Calculate the expected value of the likelihood function

$$Q\left(\theta \mid \theta^{(t)}\right) = E_{\mathbf{s}_{i_{l_k}}|\mathbf{y}_{q_i};\theta^{(t)}}\left[\log p\left(\mathbf{y}_{q_i}, \mathbf{s}_{i_{l_k}}; \theta^{(t)}\right)\right]$$
$$+E_{\mathbf{s}_{i_{l_k}}|\mathbf{y}_{q_i};\theta^{(t)}}\left[\log p\left(\mathbf{s}_{i_{l_k}}; \gamma_i, \Sigma_i\right)\right]. \tag{32}$$

M: Find the parameters $\theta = \{\sigma_{w_i}, \gamma_i, \Sigma_i\}$ that maximize

$$\theta^{(t+1)} := \arg\max_{\theta} Q\left(\theta \mid \theta^{(t)}\right). \tag{33}$$

To estimate σ_{w_i} we simplify the function in (32) by dropping the terms that do not include σ_{w_i}

$$Q\left(\sigma_{w_i} \mid \theta^{(t)}\right) \propto E_{\mathbf{s}_{i_{l_k}}|\mathbf{y}_{q_i};\theta^{(t)}}\left[\log p\left(\mathbf{y}_{q_i} \mid \mathbf{s}_{i_{l_k}}; \sigma_{w_i}\right)\right]$$
$$= -\frac{M}{2}\log\sigma_{w_i} - \frac{1}{2\sigma_{w_i}}\left\|\mathbf{y}_{q_i} - \mathbf{A}_{u_{l_k}}\mu_{s_{i_l}}\right\|_2^2$$
$$-\frac{1}{2\sigma_{w_i}}E_{\mathbf{s}_{i_{l_k}}|\mathbf{y}_{q_i};\theta^{(t)}}\left[\left\|\mathbf{A}_{u_{l_k}}\left(\mathbf{s}_{i_{l_k}} - \mu_{s_{i_l}}\right)\right\|_2^2\right] \tag{34}$$

with

$$E_{\mathbf{s}_{i_{l_k}}|\mathbf{y}_{q_i};\theta^{(t)}}\left[\left\|\mathbf{A}_{u_{l_k}}\left(\mathbf{s}_{i_{l_k}} - \mu_{s_{i_l}}\right)\right\|_2^2\right]$$
$$= Tr\left(\mathbf{C}_i\mathbf{A}_{u_{l_k}}^T\mathbf{A}_{u_{l_k}}\right) \tag{35}$$
$$= \sigma_{w_i}^{(t)}Tr\left(\mathbf{C}_i\left(\mathbf{C}_i^{-1} - \mathbf{C}_{i_0}^{-1}\right)\right) \tag{36}$$
$$= \sigma_{w_i}^{(t)}\left(k - Tr\left(\mathbf{C}_i\mathbf{C}_{i_0}^{-1}\right)\right) \tag{37}$$

where (36) can be directly derived by substituting (29) in (35). The learning rule for σ_{w_i} is obtained by setting the derivative of

$Q(\sigma_{w_i}|\theta^{(t)})$ to zero leading to

$$\sigma_{w_i}^{(t+1)} = \frac{\left\|\mathbf{y}_{q_i} - \mathbf{A}_{u_{l_k}}\mu_{s_{i_l}}\right\|_2^2 + \sigma_{w_i}^{(t)}\left(k - Tr\left(\mathbf{C}_i\mathbf{C}_{i_0}^{-1}\right)\right)}{M}.$$

To estimate γ_i and Σ_i we notice that the first term in (32) is unrelated to the aforementioned parameters and thus it can be simplified to

$$Q\left(\gamma_i, \Sigma_i \mid \theta^{(t)}\right) \propto E_{\mathbf{s}_{i_{l_k}}|\mathbf{y}_{q_i};\theta^{(t)}}\left[\log p\left(\mathbf{s}_{i_{l_k}}; \gamma_i, \Sigma_i\right)\right] \quad (38)$$

where it can be easily shown that $\log p(\mathbf{s}_{i_{l_k}}; \gamma_i, \Sigma_i) \propto -\frac{k}{2}\log|\gamma_i\Sigma_i| - \frac{\mathbf{s}_{i_{l_k}}^T(\gamma_i\Sigma_i)^{-1}\mathbf{s}_{i_{l_k}}}{2}$. Therefore (38) may be written as

$$Q\left(\gamma_i, \Sigma_i \mid \theta^{(t)}\right) \propto -\frac{k}{2}\log|\gamma_i| - \frac{k}{2}\log|\Sigma_i| - \frac{Tr\left((\gamma_i\Sigma_i)^{-1}\left(\mathbf{C}_i + \mu_{s_{i_l}}\mu_{s_{i_l}}^T\right)\right)}{2} \quad (39)$$

where $\mu_{s_{i_l}}$, $\mathbf{C}_i$ are evaluated according to (12) and (13) respectively. The derivative of (39) with respect to γ_i is given by $\frac{\partial Q(\gamma_i,\Sigma_i|\theta^{(t)})}{\partial\gamma_i} = -\frac{k}{2\gamma_i} - \frac{1}{2\gamma_i^2}Tr(\Sigma_i^{-1}(\mathbf{C}_i + \mu_{s_{i_l}}\mu_{s_{i_l}}^T))$, and the learning rule for γ_i will be $\gamma_i = \frac{Tr(\Sigma_i^{-1}(\mathbf{C}_i + \mu_{s_{i_l}}\mu_{s_{i_l}}^T))}{k}$.

While the gradient of (39) with respect to Σ_i will be $\frac{\partial Q(\gamma_i,\Sigma_i|\theta^{(t)})}{\partial\Sigma_i} = -\frac{k\Sigma_i^{-1}}{2} + \frac{\Sigma_i^{-1}(\mathbf{C}_i + \mu_{s_{i_l}}\mu_{s_{i_l}}^T)\Sigma_i^{-1}}{2\gamma_i}$. By setting the derivative to zero we obtain the learning rule for Σ_i. ■

C. Proof of Proposition 3

Proof: The definition of RIP states that, for a matrix $\tilde{\mathbf{A}} \in \Re^{M\times n_{l_e}}$ $(M < n_{l_e})$ with ℓ_2-normalized columns and for an integer scalar $s \leq M$, given that there is a constant $\rho_s \in (0,1)$, which represents the smallest quantity such that $\forall\mathbf{x} \in \Re^{s\times 1}$, then the expression

$$(1-\rho_s)\|\mathbf{x}\|^2 \leq \|\tilde{\mathbf{A}}_{(:,1:s)}\mathbf{x}\|^2 \leq (1+\rho_s)\|\mathbf{x}\|^2 \quad (40)$$

holds true for any choice of s columns. Then $\tilde{\mathbf{A}}$ is said to have an s-RIP property with a constant ρ_s. In the following we will prove that there is such constant with $\rho_s \in (0,1)$. First, note that for any vector $\mathbf{x}$ we have that

$$\|\tilde{\mathbf{A}}\mathbf{x}\|^2 = \frac{1}{\sqrt{M}}\|\mathbf{A}'_{u_{l_k}}\mathbf{x}\|^2 = \frac{1}{\sqrt{M}}\|\mathbf{A}_{u_{l_k}}\mathbf{x}\|^2 + \frac{1}{\sqrt{M}}\|\mathbf{L}_{u_{l_k}}\mathbf{x}\|^2. \quad (41)$$

Based on (40) and (41), the problem of finding the constant ρ_s for the matrix $\tilde{\mathbf{A}}$ can be decomposed into finding the respective constants for the matrices, $\mathbf{A}_{u_{l_k}}$ and $\mathbf{L}_{u_{l_k}}$. Considering the matrix $\frac{1}{\sqrt{M}}\mathbf{A}_{u_{l_k}}$, we know that due to its construction satisfies the k-RIP property with $\rho_k \in (0,1)$. The matrix $\mathbf{L}_{u_{l_k}} \in \Re^{M\times k}$ matrix that is formed by the first k columns of matrix $\mathbf{L}_{l_e}\mathbf{U}_{l_e}$, and thus we have that $0 \leq \|\mathbf{L}_{u_{l_k}}\mathbf{x}\|^2 \leq \|\mathbf{L}_{l_e}\mathbf{U}_{l_e}\mathbf{x}\|^2$. The eigenvalue decomposition of the Laplacian matrix is expressed as $\mathbf{L}_{l_e} = \mathbf{U}_{l_e}\mathbf{\Sigma}_{l_e}\mathbf{U}_{l_e}^T$, where $\mathbf{\Sigma}_{l_e}$ is the diagonal matrix containing the eigenvalues λ_i. Based on this, we have that

$$\|\mathbf{L}_{l_e}\mathbf{U}_{l_e}\mathbf{x}\|^2 = \mathbf{x}^T\mathbf{U}_{l_e}^T\mathbf{L}_{l_e}^T\mathbf{L}_{l_e}\mathbf{U}_{l_e}\mathbf{x} = \mathbf{x}^T\mathbf{\Sigma}^2\mathbf{x} = \|\mathbf{\Sigma}\mathbf{x}\|^2. \quad (42)$$

Hence, an upper bound can be obtained $0 \leq \delta\|\mathbf{L}_{l_e}\mathbf{U}_{l_e}\mathbf{x}\|^2 \leq \kappa\|\mathbf{x}\|^2$, where $\kappa = \lambda_{\max}^2/M$. Now, based on the Gershgorin circle theorem, the Laplacian matrix $\mathbf{L}_{l_e} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{C}$ has $\lambda_i \in [0,2]$. It follows that for $M > 4$ we have that $\lambda_i^2 \leq M < 4$. Hence, the matrix $\tilde{\mathbf{A}}$ has the s-RIP with a constant $\rho_s = \rho_k + \lambda_{\max}^2/M \in (0,1)$. ■

REFERENCES

[1] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 59–69, May 2011.

[2] W. Shi, Y. Lu, Z. Li, and J. Engelsma, "Scalable support for 3D graphics applications in cloud," in *Proc. IEEE 3rd Int. Conf. Cloud Comput.*, Jul. 2010, pp. 346–353.

[3] S. Wang and S. Dey, "Adaptive mobile cloud computing to enable rich mobile multimedia applications," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 870–883, Jun. 2013.

[4] W. Cai, V. C. M. Leung, and M. Chen, "Next generation mobile cloud gaming," in *Proc. IEEE 7th Int. Symp. Serv.-Oriented Syst. Eng.*, Mar. 2013, pp. 551–560. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6525574\nhttp://dx.doi.org/10.1109/SOSE.2013.30

[5] D. S. Alexiadis, D. Zarpalas, and P. Daras, "Real-time, full 3-D reconstruction of moving foreground objects from multiple consumer depth cameras," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 339–358, Feb. 2013.

[6] R. Mekuria, M. Sanna, E. Izquierdo, D. C. A. Bulterman, and P. Cesar, "Enabling geometry-based 3-D tele-immersion with fast mesh compression and linear rateless coding," *IEEE Trans. Multimedia*, vol. 16, no. 7, pp. 1809–1820, Nov. 2014.

[7] D. b. Chen, D. b. Cohen-Or, O. b. Sorkine, and S. b. Toledo, "Algebraic analysis of high-pass quantization," *ACM Trans. Graph.*, vol. 24, no. 4, pp. 1259–1282, 2005. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-33645780429&partnerID=40&md5=feef429c87d039d56b4d04d90a43983a

[8] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive geometry compression," in *Proc. 27th Annu. Conf. Comput. Graph. Interactive Technol.*, 2000, pp. 271–278. [Online]. Available: http://dl.acm.org/citation.cfm?id=344922

[9] P. Alliez and M. Desbrun, "Valence-driven connectivity encoding for 3D Meshes," *Comput. Graph. Forum*, vol. 20, no. 3, pp. 480–489, 2001. [Online]. Available: http://doi.wiley.com/10.1111/1467-8659.00541

[10] K. Kumar and Y. H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?," *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[11] A. Maglo, G. Lavoué, F. Dupont, and C. Hudelot, "3D mesh compression: Survey, comparisons, and emerging trends," *ACM Comput. Surveys*, vol. 47, no. 3, 2015, Art. no. 44. [Online]. Available: http://search.proquest.com/docview/1676467871?accountid=15533

[12] M. Deering, "Geometry compression," in *Proc. 22nd Annu. Conf. Comput. Graph. Interactive Technol.*, 1995, pp. 13–20. [Online]. Available: http://dl.acm.org/citation.cfm?id=218391

[13] G. Taubin and J. Rossignac, "Geometric compression through topological surgery," *ACM Trans. Graph.*, vol. 17, no. 2, pp. 84–115, Apr. 1998. [Online]. Available: http://portal.acm.org/citation.cfm?doid=274363.274365

[14] S. Gumhold and W. Straßer, "Real time compression of triangle mesh connectivity," in *Proc. 25th Annu. Conf. Comput. Graph. Interactive Technol.*, 1998, pp. 133–140. [Online]. Available: http://w3imagis.imag.fr/Membres/Frederic.Durand/Book/sig98.html

[15] J. Rossignac, "Edgebreaker: Connectivity compression for triangle meshes," *IEEE Trans. Vis. Comput. Graph.*, vol. 5, no. 1, pp. 47–61, Jan. 1999.

[16] C. Touma and C. Gotsman, "Triangle mesh compression," in *Graph. Interface*, 1998, pp. 26–34. [Online]. Available: http://www.Graphinterface.org/proceedings/1998/107/paper107.ps.gz

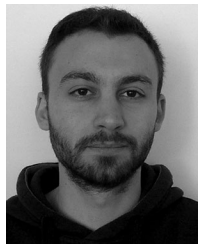
[17] K. Mamou, T. Zaharia, and F. Prýteux, "A triangle-fan-based approach for low complexity 3D mesh compression," in *Proc. Int. Conf. Image Process.*, 2009, pp. 3513–3516.

[18] O. Sorkine, "Laplacian mesh processing," in *Proc. Eurographics - State Art Rep.*, 2005, pp. 53–70.
[19] Z. Karni and C. Gotsman, "Spectral compression of mesh geometry," in *Proc. 27th Annu. Conf. Comput. Graph. Interactive Technol.*, 2000, pp. 279–286. [Online]. Available: http://portal.acm.org/citation.cfm?doid=344779.344924
[20] H. Zhang, O. Van Kaick, and R. Dyer, "Spectral mesh processing," *Comput. Graph. Forum*, vol. 29, no. 6, pp. 1865–1894, 2010.
[21] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006. [Online]. Available: http://ieeexplore.ieee.org/ielx5/18/33885/01614066.pdf?tp=&arnumber=1614066&isnumber=33885
[22] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
[23] S. Pudlewski and T. Melodia, "Compressive video streaming: Design and rate-energy-distortion analysis," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 2072–2086, Dec. 2013.
[24] C. Deng, W. Lin, B. Lee, and T. Lau, "Robust image coding based upon compressive sensing," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 278–290, Apr. 2012.
[25] Y. Rachlin and R. D. Baron, "The secrecy of compressed sensing measurements," in *Proc. 46th Annu. Allerton Conf. Commun., Control, Comput.*, 2008, pp. 813–817.
[26] A. M. Abdulghani and E. Rodriguez-Villegas, "Compressive sensing: From 'compressing while sampling' to 'compressing and securing while sampling'," in *Proc. Annu. Int. Conf. IEEE Eng. Medi. Biol. Soc.*, Aug.-Sep. 2010, pp. 1127–1130.
[27] D. Zhuo-ming and G. Guo-hua, "3-D geometric signal compression method based on compressed sensing," in *Proc. Int. Conf. Electron., Commun. Control*, 2011, pp. 601–604. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6066540
[28] F. Cayre, P. Rondao-Alface, F. Schmitt, B. Macq, and H. Maître, "Application of spectral decomposition to compression and watermarking of 3D triangle mesh geometry," *Signal Process., Image Commun.*, vol. 18, no. 4, pp. 309–319, 2003.
[29] B. Levy and H. Zhang, "Spectral mesh processing," *ACM SIGGRAPH 2010 Course*, 2010, Art. no. 8. [Online]. Available: http://www.emu.edu/cjp/spi/courses/624_Peacebuilding_Monitoring__Evaluation_and_Learning_Draft.pdf
[30] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, 1998. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/S1064827595287997
[31] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun. 2010.
[32] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

Aris S. Lalos (S'07–M'08) received the Diploma, M.A.Sc., and Ph.D. degrees from the Computer Engineering and Informatics Department (CEID), University of Patras (UoP), Patras, Greece, in 2003, 2005, and 2010, respectively.

He has been a Research Fellow with the Signal Processing and Communications Laboratory, CEID, UoP, from 2005 to 2010, and with the Signal Theory and Communications Department, Technical University of Catalonia, Barcelona, Spain, from October 2012 to December 2014. From October 2011 to October 2012, he was a Telecommunication Research Engineer with Analogies S.A., Rio-Patras, Greece. He is currently a Post-Doctoral Research Fellow with the Visualization and Virtual Reality Group, UoP. He has authored 44 research papers in 19 international journals and 22 conferences, and 3 edited books. He has participated in several European projects related to the ICT and eHealth domain (e.g., COOPCOM, ALPHA, WSN4QoL, KinOptim, MOMIRAS, MyAirCoach, etc.). His current research interests include digital communications, adaptive filtering algorithms, geometry processing, wireless body area networks, and biomedical signal processing.

Mr. Lalos acts as a Regular Reviewer for several technical journals. He was the recipient of the Best Demo Award in IEEE CAMAD 2014, the Best Paper Award in IEEE ISSPIT 2015, and was nominated as Exemplary Reviewer for the IEEE COMMUNICATIONS LETTERS in January 2015.



Iason Nikolas (S'11) received the Diploma degree in electrical and computer engineering from the University of Patras, Patras, Greece, in 2014, where he is currently working toward the Ph.D. degree.

His research interests include information and scientific visualization, mesh compression, and computer graphics.

Mr. Nikolas is a Member of the IEEE Computer Society, the IEEE EMB, Greece chapter, and the Visualization and Virtual Reality Group at the University of Patras since February 2014.

Evangelos Vlachos (S'11–M'13) received the Diploma, M.Sc., and Ph.D. degrees in signal processing for wireless communications from the Computer Engineering and Informatics Department, University of Patras, Patras, Greece in, 2005, 2009, and 2015, respectively.

He has worked as a Researcher in several European and national projects related to ICT and power grids (e.g., ENDECON, GRIC, THERMOCAMERA, and HANDiCAMS). His research interests include distributed estimation algorithms for wireless sensor networks, privacy-preserving techniques for distributed networks, optimal sensor selection, blind distributed beamforming techniques, and adaptive ICI mitigation for vehicular communications.

Konstantinos Moustakas (M'07) 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 served as a Post-Doctoral Research Fellow with the Information Technologies Institute, Centre for Research and Technology Hellas, Hellas, Greece. He is currently an Associate Professor with the Electrical and Computer Engineering Department, University of Patras, Patras, Greece, and Head of the Visualization and Virtual Reality Group, University of Patras. He has authored or coauthored more than 120 papers in refereed journals, edited books, and international conferences. His main research interests include virtual, augmented, and mixed reality, 3D geometry processing, haptics, virtual physiological human modeling, information visualization, physics-based simulations, computational geometry, computer vision, and stereoscopic image processing.

Prof. Moustakas is a Member of the IEEE Computer Society and Eurographics. He serves as a Regular Reviewer for several technical journals and has participated in more than 17 research and development projects funded by the EC and the Greek Secretariat of Research and Technology. He serves as the Coordinator or the Scientific Coordinator in four of them. He has also been a Member of the organizing committee of three international conferences and Member of the Technical Program Committee for more than 15 international conferences.