Sparse Coding of Dense 3d Meshes in Mobile Cloud Applications

Aris S. Lalos, Iason Nikolas and Konstantinos Moustakas
Department of Electrical and Computer Engineering
University of Patras, Greece
Emails: {aris.lalos, iason.nikolas, moustakas}@ece.upatras.gr

***Abstract*—With the growing demand for easy and reliable generation of 3D models representing real-world objects and environments in mobile cloud computing platforms, new schemes for acquisition, storage and transmission of 3D meshes are required. In general, 3D meshes consist of two distinct components: 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 energy efficient geometry compression system can be achieved by: i) reducing the amount of data that should be transmitted ii) minimizing the computational operations executed at the encoder. In this paper, we propose a Bayesian learning approach that allows processing large meshes in parts and reconstructing the Cartesian coordinates of each part from a small number of random linear combinations. The proposed compression/reconstruction approaches minimize the samples that are required for transmission yet assuring accurate reconstruction at the receiver, by exploiting specific local characteristics of the surface geometry in the graph Fourier domain. Simulation studies show that the proposed schemes, as compared to the state of the art approaches, achieve competitive Compression Ratios (CRs), offering at the same time significantly lower compression computational complexity, which is essential for mobile cloud computing platforms.**

I. Introduction

Recently, there has been increasing interest from researchers, system designers, and application developers on 3D modeling of real-world objects and environments, in mobile cloud computing (MCC) platforms [1], [2]. The evolution of MCC platforms and the improved mobility and interactivity of modern smart-phones and tablet computers, open up new possibilities for live acquisition, transmission and processing of 3D models to the cloud. The most common way of representing 3D models in graphic 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 the vertex connectivity. It is worth mentioning that the encoded geometry is on average more that five times larger that the encoded connectivity [3], since the raw geometry data, whether originating from scanned real-world objects or synthetic modeling applications, are represented using floating point precision. As a result, although state-of-the-art connectivity encoders are extremely effective [4], [5], the compression of geometry information not only seems to remain a challenge [3] but also, becomes more essential in MCC settings [1] 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

The geometry compression efficiency of the algorithms running on mobile devices can be optimized by proposing encoding schemes, with high Compression Ratio (CR) capabilities and reduced computational requirements. However, the vast majority of the schemes available in the literature [6]–[8] charge the transmitter with most of the processing, thus not coping effectively with these requirements. To be more specific the aforementioned works, propose the computation of the eigenvectors of the Laplacian of the mesh at the transmitter, exploiting the fact that 3D models can be well approximated by a combination of low-frequency Laplacian eigenvectors, also known as graph fourier basis vectors. The main drawback of the aforementioned schemes, is the increased processing demands at the transmitter, since they require the computation of the Laplacian eigenvectors 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 signal compression/reconstruction, providing a systematic approach for reconstructing sparse signals from a small number of random linear observations [9], [10]. More importantly, these schemes allow the progressive compression of 3D oblects, where an early, coarse approximation can subsequently be improved by simply transmitting additional random linear combinations. The authors in [11], exploited the sparse structure of the Laplacian of a 3D mesh in the eigen-domain, by employing conventional compressed sensing (CS) schemes. However, as the number of vertices of the 3D model grows, the proposed compression/reconstruction methods becomes infeasible, since they require the inversion of matrices, with sizes that are equal to the number of mesh vertices. To overcome this limitation the object should be divided and processed in submeshes.

In this paper, motivated by the aforementioned open issues, we introduce a novel geometry compression/reconstruction algorithm that enhances the benefits of the state of the art schemes, by taking into account during reconstruction, specific local characteristics (e.g., potential correlations, block sparsity) of the geometric information in the graph Fourier

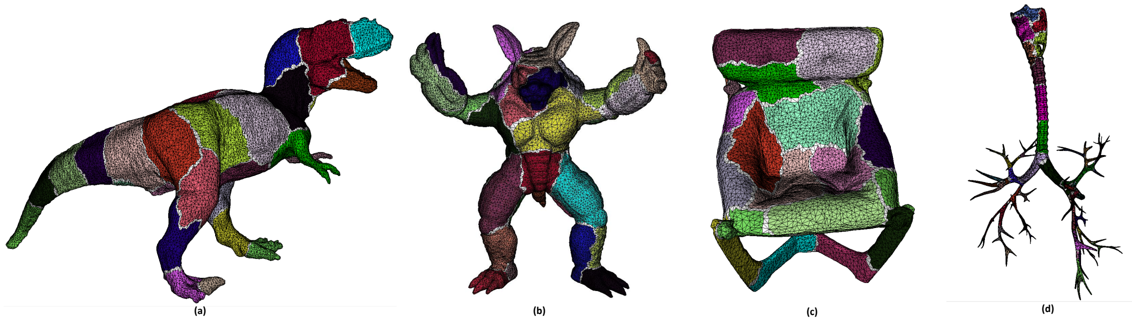


Fig. 1. *Graph partitioning using MeTiS. Submeshes are colored with randomly selected colors. Triangles with vertices that belong to different submeshes are colored white (a) Stanford Tyrannosaurus Model,* 20002 *vertices,* 60000 *edges,* 30 *submeshes (b) Stanford Armadillo Model,* 20002 *vertices,* 60000 *edges,* 30 *submeshes (c) A chair ,* 15340 *vertices,* 45990 *edges,* 30 *submeshes* (d) A lung 3d model re-constructed from CT scans, 25992 vertices and 79185 edges, 50 submeshes.

domain. The contribution of this paper can be summarized as follows:

- We propose a novel 3D model encoding/decoding architecture for MCC platforms, which decompose large meshes using a fast graph partitioning method, and compresses the geometric information of overlapped submeshes *by performing only additions* at the encoder.
- We present a novel reconstruction scheme that exploits the exponentially decaying structure of the euclidean coordinates in the graph fourier domain through the use of a modeled based Bayesian learning framework. Simulation studies carried out with different 3D objects, show that the proposed approaches, as compared to the state of the art schemes, achieve competitive Compression Ratios (CRs) offering at the same time significantly lower compression computational complexity, which is essential for MCC platforms.

The remainder of this paper is organized as follows. In Section II, we briefly review concepts and terminology related to the graph fourier based compression. In Section III, we present the proposed encoding architecture. In Section IV, we present the developed reconstruction algorithm. In Section V, 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 VI concludes this paper.

B. 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;

II. 3D Mesh Compression using Graph Fourier Transform

In this work we focus on triangle meshes, since they 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., [6], [7], [12] 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:

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

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

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

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

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

where Σ is a diagonal matrix consisting of the eigenvalues of Λ 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., x) to the eigenvectors $\mathbf{U}$ of the Laplacian operator $\mathbf{L}$ results in sparse representations $\mathbf{s}^x = \mathbf{U}^T\mathbf{x}$ that allows the following compact representation:

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

III. Proposed Architecture

In this section, we describe the operations performed for compressing the raw geometry data, while we assume that the connectivity information is available at the decoder. Our goal is to propose compression/reconstruction schemes that minimize the processing at the transmitter side without reducing the reconstruction quality at the destination. To than end, we propose a solution based on the CS framework that allows reconstruction from a small number of linearly combined points,

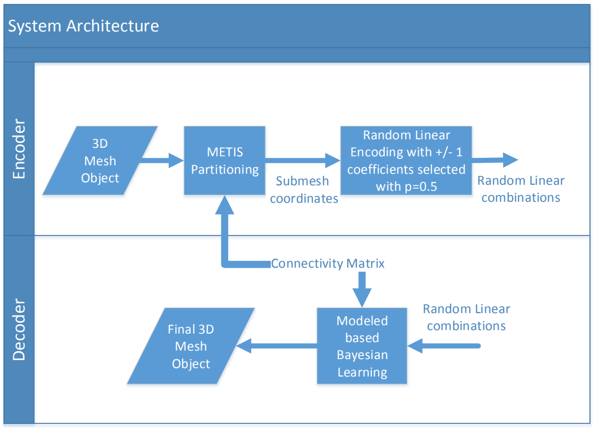


Fig. 2. *Encoder and Decoder Architecture running on a mobile device and the cloud respectively.*

by exploiting the the sparsity of the mesh point coordinates in the GFT domain. The evaluation of the corresponding transformation matrix, requires the estimation of the eigenvectors of a matrix with a size equal to the number of vertices in the mesh. The computational inefficiency of this operation, (e.g., when the number of vertices is more than 1,000) motivated the partitioning of the mesh into submeshes, that are individually processed (e.g., compressed at the encoder and recosntructed at the decoder) [7]. Figure 2 illustrates a block diagram of the system under study, while the compression and reconstruction operations are described in the following subsections.

A. Partitioning and processing of submeshes at the encoder

In the encoder side, the original 3D mesh is divided into L submeshes by using the METIS method described in [13]. Each submesh l consist of n_l nodes, where $\sum_{i=1}^{L} n_i = n$. The Cartesian coordinates of the n_l nodes included in the l-th submesh are 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 1 illustrates partitions generated by MeTiS for the Armadillo, Tyrannosaurus and a Chair model.

The trade-off included in not treating the whole mesh as one, is the possible degradation in the reconstruction quality along the submesh boundaries, so called as "edge effects". 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 overlapped submeshes, by extending each submesh with the neighbors of the boundary nodes of adjacent submeshes. To be more specific, if $I_l = [I_1, \dots, I_{n_l}]$ denotes the set of indices of the 3D mesh nodes belonging in submesh l and $I_{l_b} = \{i_{l_1}, \dots, i_{l_b}\} \subset I_l$ 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 $\cup_{j=1}^{b} N(i_{l_j}) \setminus I_l$. The extended vector will be denoted by v_{l_e} and will consist of $n_{l_e} > n_l$ vertices.

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

$$\begin{aligned} \mathbf{y}_q &= Q\left(\mathbf{A}\mathbf{v}_{l_e}\right) = \mathbf{A}\mathbf{v}_{l_e} + \mathbf{w}_q, && (5) \\ &= \mathbf{A}_{u_l}\mathbf{s}_{l_e} + \mathbf{w}_q && (6) \end{aligned}$$

where $\mathbf{A}_{u_l} = \mathbf{A}\mathbf{U}_{\mathbf{l_e}}$, $\mathbf{U}_{\mathbf{l_e}}$ are the eigenvalues of the Laplacian operator $\mathbf{L}_{l_e}$ of the extended submesh l, $\mathbf{s}_{l_e} \in \Re^{n_{l_e} \times 3}$ are the projected Cartesian coordinated in the corresponding graph fourier basis and $Q : \Re \rightarrow \mathbf{Y_i}$ is a scalar quantization function[3] that discretizes its input, by performing a mapping of each real element of $\mathbf{y}$ to a finite set of codewords $\mathbf{Y_i}$ and $\mathbf{w}_q$ represents the quantization error. The encoded samples are then transmitted to the receiver where the reconstruction of the original 3D object is taking place. At this point it should be mentioned that the overhead introduced by the transmission of the random encoding coefficients, can be significantly reduced by adopting the following policy: instead of transmitting a full encoding matrix, we transmit the first row and then the generation of the $M-1$ rows at the decoder side, by performing predefined shifts of the received row. In addition, we used the same encoding coefficients for compressing all the object submeshes. In the case that a submesh consist of more vertices than the previous one the extra encoding coefficients are generated by cyclicly shifting the already available ones. Experimental results have shown that the aforementioned strategies do not affect the performance of the decoding algorithms presented in Section . Thus, it is reasonable to neglect the communication overhead that is introduced by the transmission of the encoding coefficients and assume that matrix $\mathbf{A}$ is considered to be known at the decoder.

B. Reconstruction via Modeled 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 and ii) the DCT coefficient values of natural images and audio signal are usually modeled as multivariate gaussian distributions, 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:

$$\mathbf{s}^{\mathbf{x}}_{\mathbf{l_e}} = \left[\mathbf{s}^{\mathbf{x}}_{\mathbf{l_k}}, \mathbf{0}_{n_{l_e}-k}\right]^T, \quad \mathbf{s}^{\mathbf{x}}_{\mathbf{l_k}} = \left[s^x_{l_1}, \dots, s^x_{l_k}\right], k < n_{l_e} \qquad (7)$$

where $\mathbf{s}^{\mathbf{x}}_{\mathbf{l_k}}$ denotes the non zero block of size k that can be modeled as a parametrized multivariate Gaussian distribution:

[1]The value of M determines the achieved compression ratio as it will be shown in Section IV.

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

[3]Typical quantizers are usually optimized by selecting decision boundaries and output levels in order to minimize the distortion (e.g., mean square error) between the input real number and its quantized representation.

$$p\left(\mathbf{s}_{\mathbf{l_k}}^{\mathbf{x}}\right) \sim N\left(0, \mathbf{C}_0\right), \quad \mathbf{C}_0 = \gamma_0 \Sigma_x, \tag{8}$$

where γ_0 is a scalar parameter and $\Sigma_x \in \Re^{k \times k}$ is a positive definite matrix. By using the Bayes rule and assuming that the noise vector in $\mathbf{w}_q$ in (6) consist of Gaussian i.i.d. random variables $\mathbf{w}_q \sim N(0, \sigma_w \mathbf{I}_M)$ we obtain the posterior density of $\mathbf{s}_{\mathbf{l_k}}^{\mathbf{x}}$, which is also Gaussian $p\left(\mathbf{s}_{\mathbf{l_k}}^{\mathbf{x}} \mid \mathbf{y}_{q_x}; \sigma_w, \gamma_0 \Sigma_x\right) \sim N\left(\mu_{s_l^x}, \mathbf{C}_x\right)$ with the following mean and covariance matrix:

$$\mu_{s_l^x} = \mathbf{C}_0 \mathbf{A}_{u_{l_k}}^T \left(\mathbf{A}_{u_{l_k}} \mathbf{C}_0 \mathbf{A}_{u_{l_k}}^T + \sigma_w \mathbf{I}_M\right)^{-1} \mathbf{y}_q^x \tag{9}$$

$$\mathbf{C}_x = \mathbf{C}_0 - \mathbf{C}_0 \mathbf{A}_{u_{l_k}}^T \left(\mathbf{A}_{u_{l_k}} \mathbf{C}_0 \mathbf{A}_{u_{l_k}}^T + \sigma_w \mathbf{I}_M\right)^{-1} \mathbf{A}_{u_{l_k}} \mathbf{C}_0 \tag{10}$$

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 $\mathbf{A}_{u_l}$. Thus, given the parameters $\sigma_w, \gamma_0, \Sigma_x$ the maximum a posteriori (MAP) estimate of the Cartesian coordinates of the extended submesh l is given by

$$\hat{\mathbf{v}}_{l_e} = \mathbf{U}_{\mathbf{l_e}} \begin{bmatrix} \mu_{s_l^x} & \mu_{s_l^y} & \mu_{s_l^z} \\ \mathbf{0}_{n_{l_e}-k} & \mathbf{0}_{n_{l_e}-k} & \mathbf{0}_{n_{l_e}-k} \end{bmatrix} \tag{11}$$

To find the parameters $\sigma_w, \gamma_0, \Sigma_i$ we employ the expectation maximization (EM) algorithm to maximize $p\left(y_q^i; \sigma_w, \gamma_0, \Sigma_i\right)$ per coordinate, meaning that i can be x, y or z:

$$\sigma_w = \frac{\left\|\mathbf{y}_q - \mathbf{A}_{u_l} \mathbf{s}_{\mathbf{l_e}}\right\|_2^2 + \sigma_w \left[k - Tr\left(\mathbf{C}_x \mathbf{C}_0^{-1}\right)\right]}{M} \tag{12}$$

$$\gamma_0 = \frac{Tr\left(\Sigma_i^{-1}\left(\mathbf{C}_x + \mu_{s_l^i}\left(\mu_{s_l^i}\right)^T\right)\right)}{k} \tag{13}$$

$$\Sigma_i = \frac{\mathbf{C}_x + \mu_{s_l^i}\left(\mu_{s_l^i}\right)^T}{\gamma_0} \tag{14}$$

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

$$r = sign\left(m_1/m_0\right) min\left\{\left|m_1/m_0\right|, 0.99\right\} \tag{15}$$

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 algorithm is summarized in Table I.

IV. Simulation Results

The focus of this study is: i) to evaluate the benefits of processing overlapped submeshes as compared to the non overlapping case ii) to identify the benefits of the proposed compression/reconstruction schemes as compared to the traditional compression approaches. The proposed schemes are studied by using different 3D Objects.

TABLE I

Mesh Reconstruction via Modeled based 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 $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \Re^{n \times 1}$

For each Submesh $l = 1, \ldots, L$,
  For each iteration $m = 1, \ldots, K$

   a. Evaluate non zero values $\mu_{s_l^x}, \mu_{s_l^y}, \mu_{s_l^z}$ from (9)

   b. Evaluate the corresponding variances $\mathbf{C}_x, \mathbf{C}_y, \mathbf{C}_z$ via (10)

   c. Update $\sigma_w, \gamma_0, \Sigma_i$ for every coordinate i=x,y,z from (12)-(14)

   d. Update the value of r from (15) and re-evaluate
   $\Sigma_i = Toeplitz\left\{\left[1 \ r \ \ldots r^{k-1}\right]\right\}$.
  end For

  Evaluate Cartesian Coordinates $\hat{\mathbf{v}}_{l_e}$ of the l submesh from (11)
  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

A. Experimental Setup & Metrics

We assume that each object, is divided into N submeshes. Then, each submesh is compressed either by using the Conventional or the CS schemes. As conventional schemes we consider: i) the approaches that provide compact representation of 3D meshes in the GFT basis (GFT) (e.g., [7]), presented in Sec. II, ii) the approach of Sorkine and Cohen-Or [6] known as least squre meshes (LSM), that make use of specific points in the mesh so called as anchor points. The CS schemes include: i) a method similar to the one presented in [11] that reconstructs each submesh from RLCs, by exploiting the sparsity of each submesh in the GFT domain (CS GFT) ii) 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) :

$$CR = 1 - \frac{\sum_{i=1}^{L} M_i \times q}{n \times 32} \tag{16}$$

where M_i is the low frequency components (GFT Case), the generated encoded samples for submesh i (CS GFT and MBL case), or the number of Anchor points (LSM case), q is the number of bits used for the representation of the encoded samples. The quantization is performed by applying the Lloyd max algorithm [14]. The reconstruction effectiveness is evaluated by the normalized mean square visual error ($NMSVE$) defined

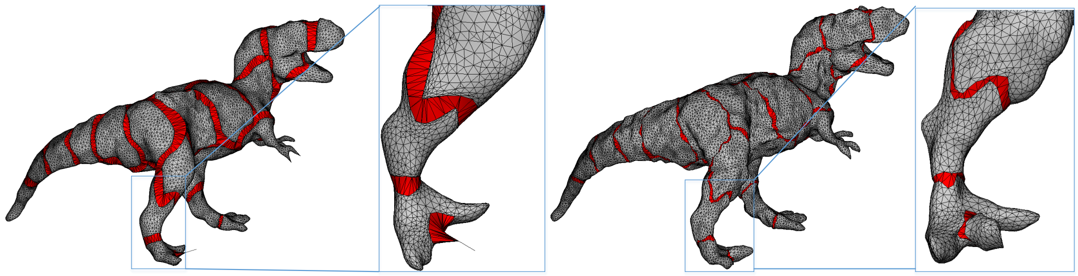


Fig. 3. *Reconstruction of the Mesh Geometry using (a) non overlapped (NMSVE: -27 dB) and (b) overlapped submeshes (NMVSE: -34 dB). The achieved CR in both cases is 0.92.*

in [7], as the average error in the Cartesian and Laplacian domains:

$$\begin{aligned} NMSVE &= \frac{1}{2n}\sum_{i=1}^{n}\left(\|\mathbf{v}_i-\tilde{\mathbf{v}}_i\|_{l_2}+\|GL(\mathbf{v}_i)-GL(\tilde{\mathbf{v}}_i)\|_{l_2}\right) \\ GL(\mathbf{v}_i) &= v_i-\frac{\sum_{j\in N(i)} d_{ij}^{-1}v_j}{\sum_{j\in N(i)} d_{ij}^{-1}} \end{aligned} \tag{17}$$

where d_{ij} denotes the Euclidean distance between i and j.

B. Performance Evaluation

To evaluate the benefits of processing overlapping submeshes, as compared to the non overlapping case presented in [7], we executed the GFT method for the same number of low frequency components per submesh. In fig. 3 we provide the reconstructed Tyranosaurus models, where it is clearly shown that the overlapping method described in section II results in a more accurate reconstruction, that is attributed to the fact that the boundary points (vertices of red triangles) are recovered almost perfectly. Based on this observation we adopt the idea of processing overlapping submeshes.

In Fig.4 (a), the obtained NMSVE for a chair model scanned by using a Kinect sensor, is plotted against the achieved CR after transmitting the $M_i = M = [65, 130, 195, 250, 280]$ lower frequency components per sub-mesh, 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 the transmitted samples were selected equal to $q = 12$. By inspecting the figure, it is clear that the application of the 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 achieves performance almost similar to the classical one (GFT). Moreover, 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 $O(n^2)$ **additions and** 0 **multiplications** for compressing the 3D object, while the GFT requires $O(n^3)$ multiplications and $O(n^3)$ additions. More importantly, the proposed method inherently involves a randomization process offering privacy preservation without any additional cost [15].

Fig. 4 (b), shows the NMSVE against the number of executed iterations for the Armadillo Model, where it is clearly shown that the MBL algorithm converges after two iterations. Finally, in fig. 3 we provide the original and reconstructed 3D objects, using the proposed approach with $K = 2$. The achieved CR is equal to 95%. By inspecting the reconstructed objects, it can be easily verified that the proposed scheme achieve high CR, offering at the same time significantly low compression computational complexity.

V. Conclusions

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 reconstruction can be obtained at the receiver, by exploiting key characteristics 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. This property is considered critical for emerging 3D model acquisition schemes on off-the-shelf mobile devices.

VI. Acknowledgments

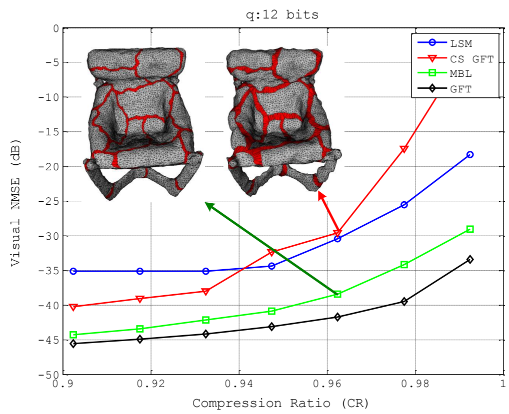
This work has been supported by the Greek Secretariat for Research and Technology Bilateral Collaboration Project MOMIRAS (*ISR*_3215) and the H2020-PHC-2014-2015 Project MyAirCoach (Grant Agreement No. 643607).

References

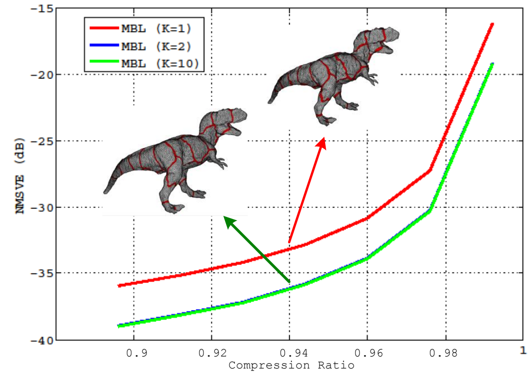
[1] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, April 2010.

[2] S. Kitanov and D. Davcev, "Mobile cloud computing environment as a support for mobile learning," in *Cloud Computing 2012, The Third International Conference on cloud computing, GRIDs, and Virtualization*, 2012, pp. 99–105.

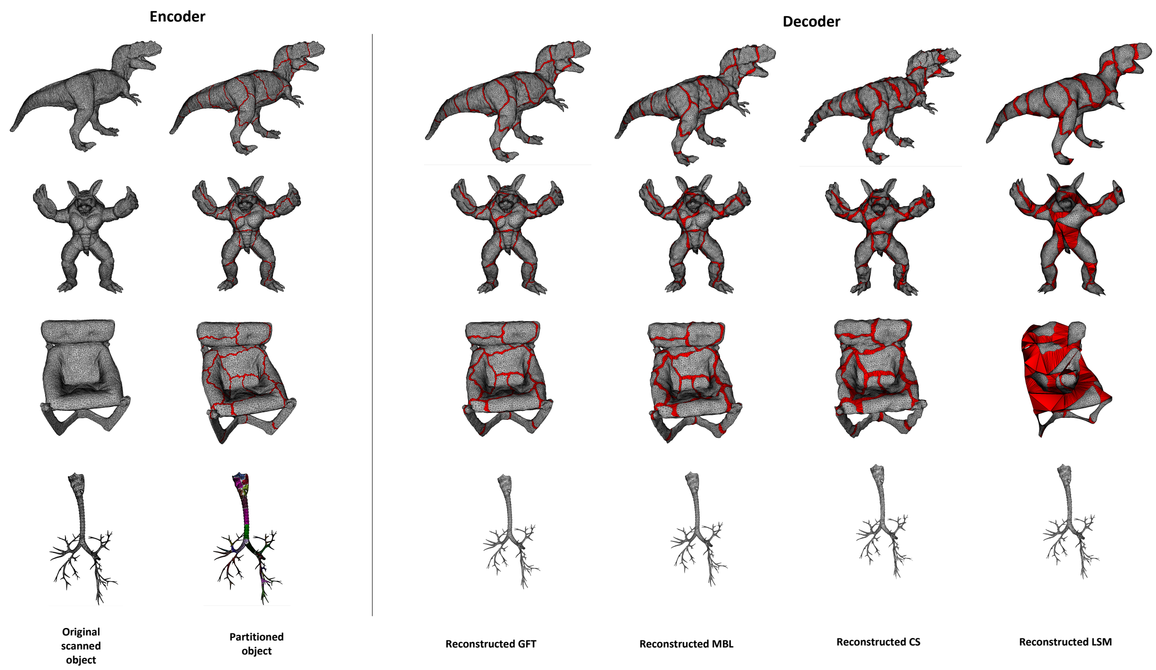
[3] D. Chen, D. Cohen-Or, O. Sorkine, and S. Toledo, "Algebraic analysis of high-pass quantization," *ACM Transactions on Graphics*, vol. 24, no. 4, pp. 1259–1282, 2005.



(a)



(b)



(c)

Fig. 4. *(a) VNMSE vs CR using the chair model, (b) VNMSE vs CR for different number of iterations (c) Original and reconstructed 3D objects, CR:0.95 .*

[4] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive geometry compression," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 271–278.

[5] P. Alliez and M. Desbrun, "Valence-driven connectivity encoding for 3d meshes." *Comput. Graph. Forum*, vol. 20, no. 3, pp. 480–489, 2001.

[6] O. Sorkine, "Laplacian mesh processing," Ph.D. dissertation, Citeseer, 2006.

[7] Z. Karni and C. Gotsman, "Spectral compression of mesh geometry," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 279–286.

[8] H. Zhang, O. Van Kaick, and R. Dyer, "Spectral mesh processing," *Computer Graphics Forum*, vol. 29, no. 6, pp. 1865–1894, 2010.

[9] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[10] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[11] Z.-M. Du and G.-H. Geng, "3-d geometric signal compression method based on compressed sensing," in *Image Analysis and Signal Processing (IASP), 2011 International Conference on*, Oct 2011, pp. 62–66.

[12] B. Lévy and H. R. Zhang, "Spectral mesh processing," in *ACM SIGGRAPH 2010 Courses*. ACM, 2010, p. 8.

[13] 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, Dec. 1998.

[14] S. Lloyd, "Least squares quantization in pcm," *IEEE Trans. Inf. Theor.*, vol. 28, no. 2, pp. 129–137, Sep. 2006.

[15] A. Abdulghani and E. Rodriguez-Villegas, "Compressive sensing: From compressing while sampling to compressing and securing while sampling," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, Aug 2010, pp. 1127–1130.