

# Compressive Representation of Three-dimensional Models

José Paulo R. de Lima, Helton Hideraldo Bísaro

Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)

josepaulolima@gmail.com, heltonhb@usp.br

**Abstract**—Due to recent developments in data acquisition mechanisms, called 3d scanners, mesh compression has become an important tool for manipulating geometric data in several areas. In this context, a recent approach to the theory of signs called Compressive Sensing states that a signal can be recovered from far fewer samples than those provided by the classical theory. In this paper, we investigate the applicability of this new theory with the purpose of to obtain a compressive representation of geometric meshes. We developed an experiment which combines sampling, compression and reconstruction of various mesh sizes. Besides figuring compression rates, we also measured the relative error between the original mesh and the recovered mesh. We also compare two measurement techniques through their processing times, which are: the use of Gaussian matrices; and the use of Noiselet matrices. Gaussian matrices performed better in terms of processing speed, with equivalent performance in compression capacity. The results indicate that compressive sensing is very useful for mesh compression showing quite comparable results with traditional mesh compression techniques.

Conventional approaches to signal theory follow the Shannon/Nyquist theorem: “the sampling rate must be at least twice the maximum frequencies in the signal”. According to Candès and Wakin [1], that result underlies almost all signal acquisition protocols used for audio and visual electronics, medical image devices, radio receivers and so on. However, most of those signals must be compressed in order to be stored. This fact suggests that the Shannon/Nyquist principle supposes the worst possible case and most of the acquired coefficients are redundant. The Compressive Sensing (CS) theory, also know as Compressive Sampling, is a paradigm that asserts that one can recover a signal or an image from far fewer samples or measures than traditional methods. The reconstruction is performed using an optimization procedure and, of course, the word “sample” has an entirely new meaning. Instead of collecting points in different regions of an  $n$ -dimensional signal  $X$  or averages in small areas, each measurement  $y_k$  in CS acquisition is the inner product of  $X$  with different test functions  $\phi_k$ .

## I. INTRODUCTION

Geometrical meshes play a very important role in areas such as computer graphics, virtual reality and numerical simulation. Meshes with high level of details can have high computational cost for storing and processing. In this paper, we provide a comparative study of mesh compressive sensing on Geometric meshes, which offer an effective way of representing 3D geometric models in a compressive representation. Meshes are obtained from several sources such as modelling software and 3D scanning.

There are several important areas of application involving meshes, for instance: finite elements simulation, where the polygonal meshes and the equation solvers are used to find numerical solution for differential equations with different boundary conditions; 3D printing; Interactive environments for Virtual Reality; Medical training simulators and even simple entertainment applications, where highly complex models are used to achieve satisfactory levels of realism. Currently, its is possible to acquire highly detailed and arbitrary topology surfaces with millions or even billions of vertices. Compressive representations on Meshes are very important in that type of scenario.

We explore, in this work, an approach that uses an acquisition paradigm named *Compressive Sensing* applied to geometric meshes. We performed an experiment that runs compressive sampling and subsequent reconstruction of geometric models of various sizes, proving that it is feasible to apply this theory in more efficient acquisition protocols for geometric models.

$$y_1 = \langle \phi_1, X \rangle, \dots, y_m = \langle \phi_m, X \rangle \quad (1)$$

In this paper we use a sparse representation of geometric information contained on meshes to obtain a compressive representation. Also, we use different types of test functions and compare the results in terms of how many “samples” need to be stored to obtain a reconstructed mesh as close as possible to the original.

## II. BASIC THEORY

In an early paper, Emmanuel J. Candès [2] suggests the possibility of new data acquisition protocols that translate analogical information into digital with less sensors than it was previously considered necessary. In the same work, the author provides the key mathematical results underlying this new theory. Candès and Wakin [3] discuss sensing mechanics in which a signal  $f(t)$  is stored using inner product with linear functions

$$y_k = \langle f(t), \phi_k(t) \rangle. \quad (2)$$

Specifically, given an  $n$ -dimensional  $k$ -sparse signal  $S$ , we aim to find an  $m \times n$  measurement matrix  $\Phi_\Omega$  and solve the optimization problem:

$$\begin{aligned} & \min \|S\|_{l_1}, \\ & \text{subject to } \Phi_\Omega S = Y \end{aligned} \quad (3)$$

The components  $y_1, \dots, y_m$  of the  $Y$  vector are called measurements of the signal  $S$ , and  $\Omega$  is a random measurement subset of size  $|\Omega| = m$ . CS theory aims to choose  $\Phi_\Omega$  to take as little measurements as possible and yet, to produce a reliable reconstruction of  $S$ .

Most times, a signal  $X$  must be rewritten as  $(\Psi X = S)$  in a proper basis in order to have a sparse representation, where  $\Psi$  is a change of basis matrix. So, instead of  $S$  in Equation 3, we have:

$$\begin{aligned} \Psi X = S &\iff \Psi^* S = X \\ \Theta_\Omega S = Y, \text{ where } \Theta_\Omega &= \Phi_\Omega \Psi^* \end{aligned}$$

The problem, as illustrated in Figure 1, of finding  $S$  such that  $\Theta_\Omega S = Y$  is ill posed since the matrix  $\Theta_\Omega$  is not invertible and not all the solutions satisfy the sparsity property. Therefore, a simple choice consists of taking, among all possible solutions, the one that makes  $S$  sparsest.

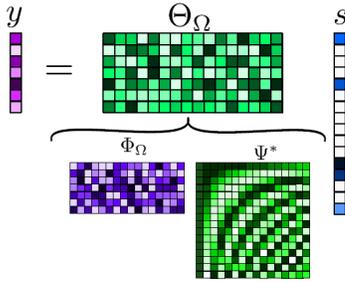


Figure 1: Illustration of the algebraic problem. Extracted from Schulz *et al.* [4]

Compressive sensing rely on two basic ideas: *sparsity* as well as *incoherence*.

**Sparsity** expresses the idea that the number of degrees of freedom of a known signal is significantly smaller than its length. Many natural signals such as images, sounds and, specially, geometrical meshes are sparse or they have a sparse representation in a proper basis  $\Psi$ .

**Incoherence** extends the duality between time and frequency. The basic idea is that signals that have a sparse representation in  $\Psi$  must be spread out of this domain. A formal definition of coherence between two bases  $\Phi$  and  $\Psi$  is:

$$\begin{aligned} \mu(\Phi, \Psi) &= \sqrt{n} \max_{j,i} |\langle \phi_j, \psi_i \rangle| \\ &\|\phi_j\|_{l_2}, \|\psi_i\|_{l_2} = 1 \end{aligned} \quad (4)$$

The definition in Equation 4 gives a measurement of the correlation between the sensing waveforms  $\phi_j$  and those where the signal is supposed to be sparse  $\psi_k$ . Notice that  $\mu(\Phi, \Psi)$  is an angle measurement between the sensing and the sparsity waveforms. Thus, high incoherence means that the vectors are nearly orthogonal. The main idea underlying this concept is that, if a random combination of the entries are measured,

something new about the sparse vector is learned in every measure.

Incoherence also can be defined in terms of the matrix  $\Theta$  as:

$$\mu(\Theta) = \max_{j,i} |\Theta_{i,j}| \quad (5)$$

Actually, this definition is equivalent to the definition given in Equation 4, since the matrix  $\Theta$  can be written as:

$$\Theta = \begin{bmatrix} \phi_1^T \\ \vdots \\ \phi_n^T \end{bmatrix} [\psi_1 \cdots \psi_n] = \begin{bmatrix} \phi_1^T \psi_1 & \cdots & \phi_1^T \psi_n \\ \vdots & \ddots & \vdots \\ \phi_n^T \psi_1 & \cdots & \phi_n^T \psi_n \end{bmatrix} \quad (6)$$

Results from CS [5], [2], [6], [7] theory establish that, in general, an image signal can be reconstructed by taking only  $O(K * \log(n))$  coefficients.

In fact, while classical sampling theory presumes an infinite length, continuous signal, the CS theory focuses on the measurements of finite vectors in  $\mathbb{R}^n$ . Another important difference is that rather than sampling a signal at a specific point in time, the CS theory “sense” the signal in terms of an inner product between the signal and some test functions. In fact, the *randomness* of these test functions play a key role in the sensing process. The third main difference between the Shannon-Nyquist framework and the CS approach is that in the first, the recovery is achieved through sinc functions interpolation, a linear procedure that uses little computation, while in the second, the recovery is achieved using highly non linear methods. An overview of those methods can be found in the work of Tropp *et al.* [8].

### III. RELATED WORK

In this section, we describe some previous work in Compressive Sensing theory and Mesh Compression.

#### A. Compressive Sensing

Medical imaging was one of the areas that benefited most from Compressive Sensing [9], [10]. In that area, processing times were speed up by a factor of seven for the same diagnostic quality. Moreover, this framework has inspired researches that extend the CS theory to practical implementations, compressive imaging and compressive sensor networks. The literature has demonstrated that data acquisition and compression can be combined to reduce the time and the space to acquire many signals of interest.

In fact, CS has made possible to perform a drastic reduction in acquisition times. Duarte *et al.* [11] have proposed a new camera architecture, which they call “single-pixel camera”. The camera combines sampling and compression in a single non adaptive acquisition process.

Patel *et al.* [12] present an algorithm called “GradientRec” that uses a CS technique to recover the horizontal and vertical gradients and then, estimates the original image from these gradients. They also present two methods to solve the inverse

problem: least-square optimization and a generalized Poisson solver. Fang and his collaborators [13] present a technique based on CS concepts for reconstruction and enhancement of multi-dimensional image data. Their method utilizes sparse representation dictionaries constructed from previously collected datasets. Lingala and Jacob [14] proposed what they called Blind Compressive Sensing frame work to to recover dynamic magnetic resonance images from undersampled measurements. Their approach use combinations of temporal basis function chosen from a large dictionary.

### B. Mesh Compression

The 3D mesh information can be divided into geometric information, parameter information, and topology or connectivity information. The geometric information concerns the vertices coordinates and the normal vectors at the vertices; the parameter information refers to any information that can be stored in the mesh, for instance, color, heat information, forces, etc.; the connectivity information is related to the neighborhood information among the vertices, the genus of the surfaces and so on. The compression of topology, in general, aims at reducing repeated references to vertices that are shared by numerous polygons in a mesh, while compression of geometrical information uses signal processing methods to represent or remove highly detailed features.

Touma and Gotsman [15] offered an algorithm, called “valence-driven approach,” that achieves a compression rate lower than 1.5 bpv on average to encode the mesh connectivity. The main drawback of this approach is that it is applicable only to orientable surfaces and manifold meshes. A similar approach, proposed by Gumhold and Strasser [16], divides the mesh into conquered and unconquered parts, inserting triangle by triangle into the conquered parts.

Rossignac [17] presents a popular example of triangle conquest approach, the Edgebreaker. In summary, this approach is a depth-first traversal of the dual graph of the mesh and it can encode an orientable manifold mesh with multiple boundaries and arbitrary genus with a guaranteed worst-case cost of 4 bpv for simple meshes. Several variations of Rossignac’s approach were developed [18], [19], [20].

Khodakovsky *et al.* [21] offer a progressive compression scheme that can handle arbitrary surface topology and highly detailed geometry. They use semi-regular wavelet transforms, zero tree coding and subdivision-based reconstruction scheme to improve their approach. Karni and Gotsman [22] project the  $x$ ,  $y$  and  $z$  coordinate vectors onto basis functions to obtain a geometric spectrum for each coordinate. Those basis functions are, in fact, the Fourier basis functions. Therefore, encoding and decoding are performed by means of the Fast Fourier Transform (FFT). The connectivity is mapped on a 6-regular connectivity without using geometry information in the process. That approach is non-optimal, but provides an acceptable trade off between performance and computational cost.

The work of Du and Geng [23] provides a compressive sensing based method for mesh compression by using a Laplace operator. According to the authors, the results are suitable for large-scale data compression, however, no experiments about bpv rates or PSNR of the results are presented.

Lobaz and Vasa [24] introduced an efficient algorithm for encoding triangle meshes based on their Laplacian coordinates that provides a global rigidity and low absolute error even for large meshes.

Maglo and co-authors [25] present an survey that summarize the early works on mesh compression and put in evidence the new approaches. They evaluate the algorithms performance and provide synthetic comparisons. The authors classified the approach to static mesh compression in three types: *Single-rate* - that build a compact representation of a mesh and, in the decompression process, produces a mesh that is either identical to the input model or only slightly different. *Progressive* - algorithms that, during decompression, reconstruct successive levels of detail as more data are decoded. *Random accessible* - methods that are used to decompress only requested parts of the input mesh in order to save resources. According to then, progressive and random accessible algorithms are not as efficient as pure single rate methods.

## IV. EXPERIMENTS AND METHODOLOGY

To apply Compressive Sensing to geometric meshes, we propose an algorithm in Matlab [26] that performs measurements which are, then, used in the reconstruction of an approximation mesh that can be compared with the original mesh.

Our algorithm decomposes the mesh into two data sets: vertices and faces. We focused our study on the mesh geometry, i.e., the set of vertices. Thus, the set of faces was kept untouched. We decompose the set of vertices into three vectors called  $X$ ,  $Y$  and  $Z$ , each of these vectors represent the data set of vertices in an axis of  $\mathbb{R}^3$ . However, the data from these vectors are not in a sparse representation. So, we need to perform a spectral decomposition using a Laplacian matrix defined as follows:

First, consider  $E$  the set of edges of a mesh  $M$  and let  $d_i$  be the number of immediate neighbours of a particular vertex  $v_i$  (the valence or degree of  $v_i$ ). Let  $A$  be the adjacency or connectivity matrix of  $M$ :

$$A_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

and let  $D$  be a diagonal matrix such that  $D_{ii} = d_i$ . Then, the Laplacian Matrix in relative coordinates is defined as:

$$L = I - D^{-1}A. \quad (8)$$

Next, we calculate the first  $k$  eigenvectors through an SVD decomposition,  $L = U\Sigma V^*$  where  $U$  is a unitary matrix  $n \times k$ ;  $\Sigma$  is a  $k \times k$  diagonal matrix and the  $k \times n$  unitary matrix  $V^*$ , denotes the conjugate transpose of  $n \times k$  matrix  $V$ . Each vector  $X$ ,  $Y$  and  $Z$  is multiplied by  $U$  to obtain a sparse representation. Figure 2 shows the result of this transformation. In Figure 2 a we can see the dense nature of the vertices’ coordinates of the mushroom model. After applying the change of basis, the coordinates become more sparse (Figure 2 b).

In our experiments, we computed  $k$  as  $C \times \log(\#(X))$ , where  $\#()$  is the number of elements of the set  $X$  and  $C$  is an empirical constant.

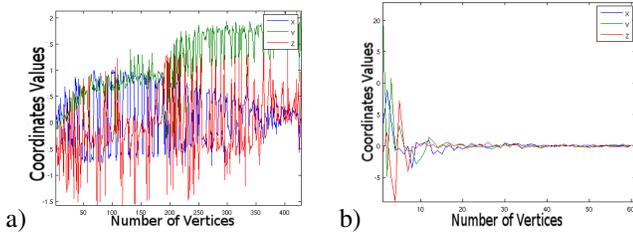


Figure 2: Plot of the spectral x Cartesian coordinates of a single mesh. a) Cartesian coordinates. b) spectral coordinates.

Then, we apply a quantization step in the spectral coordinates to obtain a sparse set of integer coordinates. This step is important because it gives an idea of the compression rate achieved by the method.

Finally, we multiply the vectors by the measurement matrix  $\Phi$ . Our experiment uses two arrays of different measurements: a Gaussian matrix and a Noiselet matrix [27]. Both matrices satisfy the incoherency property.

According to Coifman, Geshwind and Meyer [27], Noiselet matrices are constructed via a multi-scale iteration in exactly the same way as wavelet packets and have several good properties. For example, they help to distinguish that the few large values in the transformed data describe the “interesting” part of the data, and the vast majority of values, which are small, represent noise terms.

```

Data: A Mesh  $M$ 
1 Create a matrix  $O_v$  with the mesh's vertices ;
2 Create a matrix  $O_f$  with the mesh's faces (connectivity) ;
3 Set  $V_k = U * O_v$  for  $k$  meaning coefficients ;
4 Set the measurement variable  $meas$  as  $2k$  ;
5 Set the initial  $PSNR_{old}$  as 0 and the initial  $PSNR_{new}$ 
   $2 \times \epsilon$  ;
6 while  $|PSNR_{new} - PSNR_{old}| > \epsilon$  do
7    $PSNR_{old} = PSNR_{new}$  ;
8    $s = FunctionSparsity()$  ;
9    $iter = 0$ ;
10  while  $iter < \delta$  do
11    Compute the measurements matrix  $\Phi$  ;
12    Compute the vectors  $V_{m_x}, V_{m_y}$  and  $V_{m_z}$  from
       $V_k$  and  $\Phi$ ;
13    Apply the  $L_1$  norm minimization (see Equation
      3 ) and recompute  $O_v^{rec}$  ;
14    Construct  $M'$  from  $O_v^{rec}$  and  $O_f$  ;
15    Store the Hausdorff distance from  $M$  and  $M'$  in
      matrix  $D$ ;
16    Increment  $iter$  and  $s$ ;
17  Compute the  $BPV$  and  $PSRN_{new}$  ;
18  Increment  $meas$  ;
19 return  $D$ 

```

**Algorithm 1:** CS application in geometric information.

Our experiments use two variables: sparsity  $s$  and number of measurements  $meas$ . We initialize  $s$  with the value  $\frac{k}{10}$  and the increment of  $s$  also as  $\frac{k}{10}$ . The internal loop of algorithm

1 always executes 10 times (the value of  $\delta$  is set to 10). Every step of that loop, the vectors  $V_{m_x}$ ,  $V_{m_y}$  and  $V_{m_z}$  are built with only  $s$  significant coefficients.

The variable  $meas$  is initialized as  $2k$  and incremented by  $\frac{k}{4}$ . Varying both parameters, we calculate the Hausdorff distance [28] between the original mesh and the reconstructed mesh, as well as Bits per Vertex ratio  $BPV$  [29] and  $PSNR$  (Peak Signal Noise Ratio), which is an error metric widely used to compare compression strategies.

The  $PSNR$  metric produces a dimensionless measure and is less sensitive to small variations. Considering  $O_v$  a  $3 \times n$  matrix of original vertices and  $O_v^{rec}$  the set of reconstructed ones,  $PSNR$  may be defined as:

$$PSNR = 20 \log_{10} \left( \frac{\max_i |V_i|}{RMSE} \right) \quad (9)$$

where RMSE is the *root mean square error* which is computed as:

$$RMSE = \left( \frac{1}{n} \sum_{i=1}^n (O_{v_i} - O_{v_i}^{rec})^2 \right)^{\frac{1}{2}}. \quad (10)$$

In fact  $PSNR$  is dimensionless, however, because of the use of logarithm, it is usual to express it in *decibels*. According to Salomon [30], absolute values of  $PSNR$  have no meaning. The values are used only to compare different strategies. For image compression, for example, typical PSNR values are between 20 and 40.

Other authors [31] [32] also argue that  $PSNR$  is not suitable for human perception, but, as a matter of fact, other metrics would also be subjective and costly. Besides, according to Lee *et al.*[33], humans cannot recognize difference between two images when the  $PSNR$  is larger than 30-50 dB.

Figure 3 shows the meshes that were used to conduct our experiment. Mesh (a) (Mushroom) has 226 vertices and 448 faces; mesh (b) (Venus) has 711 vertices and 1,396 faces; mesh (c) (Elephant) has 2,775 vertices and 5,558 faces; mesh (d) (David's Head) has 23,889 vertices and 47,280 faces; mesh (e) (Fandisk) has 25,894 vertices and 51,784 faces and mesh (f) (Stanford Bunny) has 34,835 vertices and 69,473 faces.

All experiments were performed using a computer with the following characteristics: intel 4 x Core i5 3330S CPU @ 2.7 GHz with 8.0 Giga of RAM memory under a Ubuntu 14.04 Operating System.

## V. RESULTS AND DISCUSSION

In this section, we present the results of all mesh compression and reconstruction experiments. In order to avoid excessive repetition in the results, we display only the results of three models with the largest number of vertices. Figures 4 to 9 show the Hausdorff distances between the original meshes and those that were reconstructed using the CS technique. The distances shown in the even-numbered figures were obtained using Noiselet measurement matrices; the distances shown

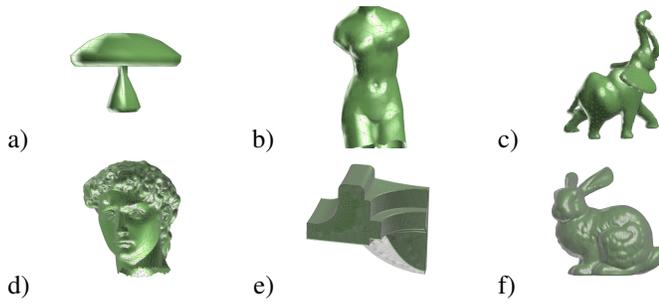


Figure 3: Meshes used in our experiment. (a) Mushroom. (b) Venus. (c) Elephant. (d) David's Head. (e) Fandisk. (f) Stanford Bunny.

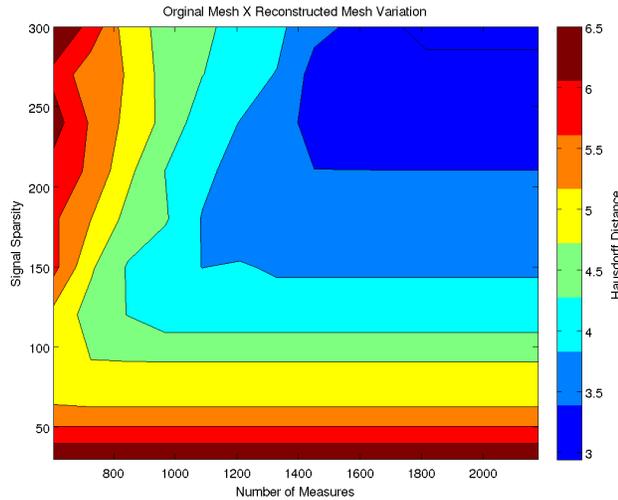


Figure 4: David's head (Hausdorff Distance between the original mesh and the reconstructed mesh): Noiselet measurement matrix. The average processing time for reconstruction was 35.2912 seconds.

in the odd-numbered figures were obtained with Gaussian measurement matrices.

In all the cases, Gaussian and Noiselet matrices produced similar results in terms of distance scale for the same mesh. The upper right corner of each graph shows the shortest distances obtained during our experiment. That area, colored in blue, displays the same magnitude for both measuring matrices tested in this work. However, only the distances are not sufficient to have a clear idea about the quality of the reconstruction, which is why we also present the Hausdorff distance depending on compression rate (bit per vertex), and the PSNR values as a function of compression ratio. In Figures 12, 14 and 16, we present the rates of compression (bits per vertex and PSNR) for three models: David's Head, Fandisk and Stanford Bunny. In all cases, we note an asymptotic behavior of the error measures. Moreover, the PSNR values are consistent with those obtained in the literature [30] and its asymptotic behavior were also consistent with the behavior observed in images, as can be seen in the work of Schulz *et al.* [34]. This trend indicates that it would be ineffective to take a larger number of measurements

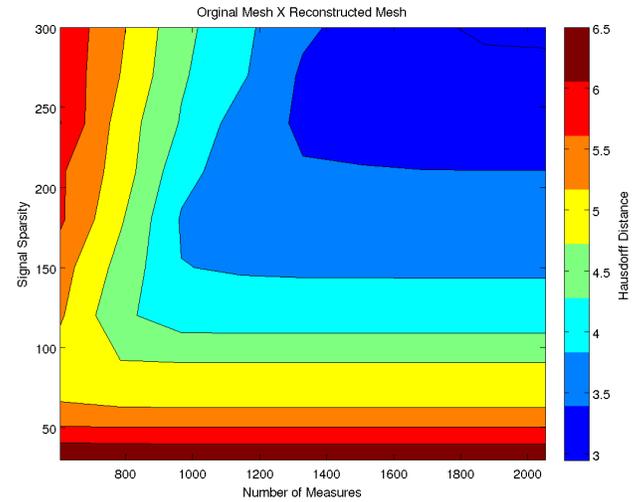


Figure 5: David's head (Hausdorff Distance between the original mesh and the reconstructed mesh): Gaussian measurement matrix. The average processing time for reconstruction was 27.2517 seconds.

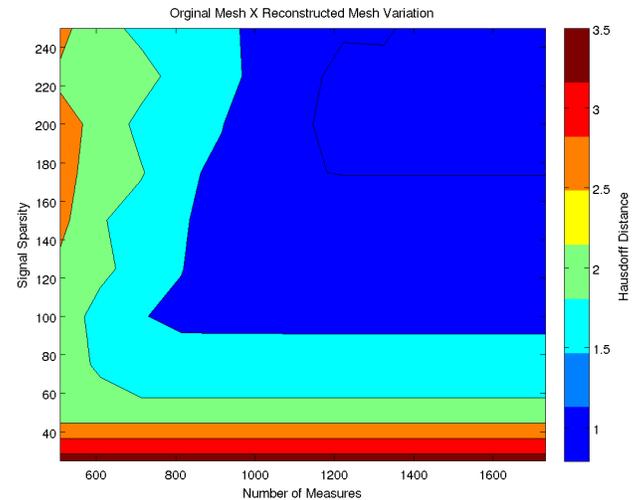


Figure 6: Fandisk (Hausdorff Distance between the original mesh and the reconstructed mesh): Noiselet measurement matrix. The average processing time for reconstruction was 42.7562 seconds.

than what has been used in our experiments.

In all cases, the compression rate, measured in bits per vertex, behaved as expected, i.e., the smaller the Hausdorff distance is, the smaller the compression rate becomes (more bits per vertex have to be used for encoding). Thus, good compression rates imply smaller numbers of bits per vertex in the encoding process. The compression ratio (bits per vertex) ranged between 1 and 3.5 BPV in the David's Head mesh (Figure 11), between 0.8 and 2.6 in Fandisk's mesh (Figure 13) and between 1 and 4 in the Stanford Bunny's mesh (Figure 15). As a benchmark, Table I shows the compression rates achieved

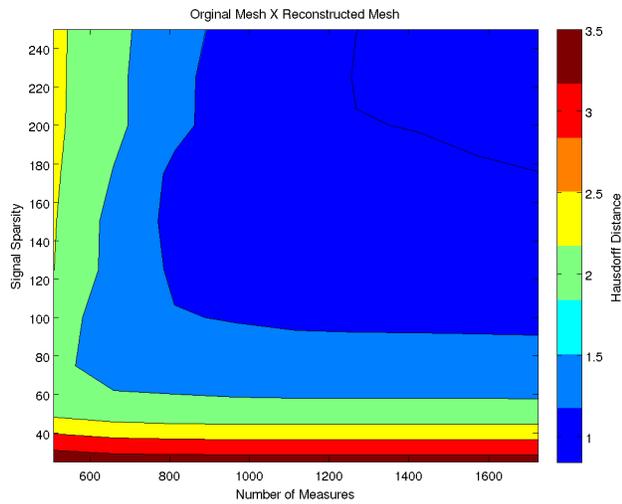


Figure 7: Fandisk (Hausdorff Distance between the original mesh and the reconstructed mesh): Gaussian measurement matrix. The average processing time for reconstruction was 27.2336 seconds.

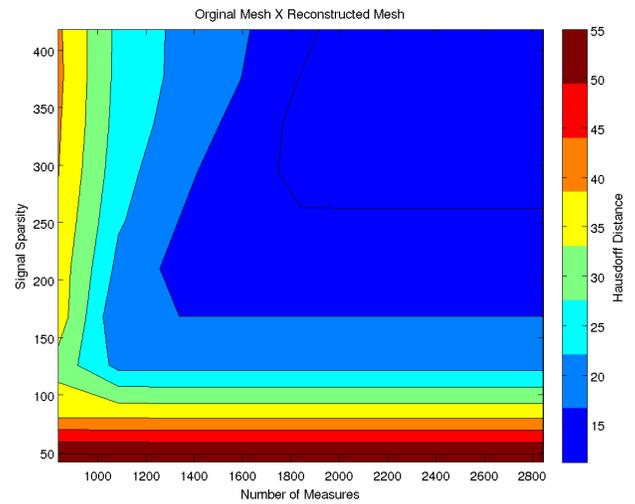


Figure 9: Stanford Bunny (Hausdorff Distance between the original mesh and the reconstructed mesh): Gaussian measurement matrix. The average processing time for reconstruction was 57.2708 seconds.

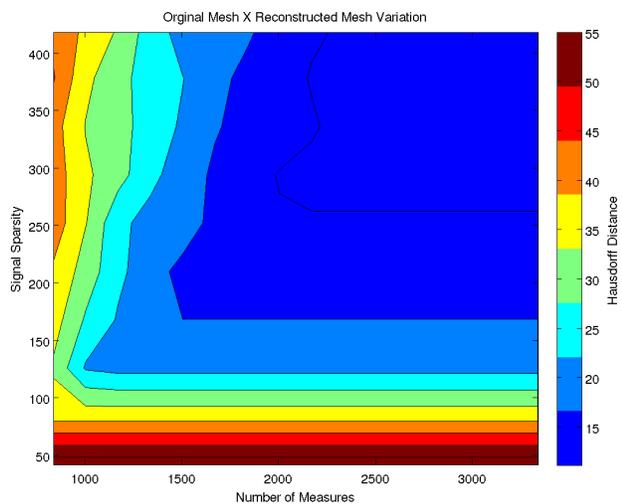


Figure 8: Stanford Bunny (Hausdorff Distance between the original mesh and the reconstructed mesh): Noiselet measurement matrix. The average processing time for reconstruction was 89.3495 seconds.

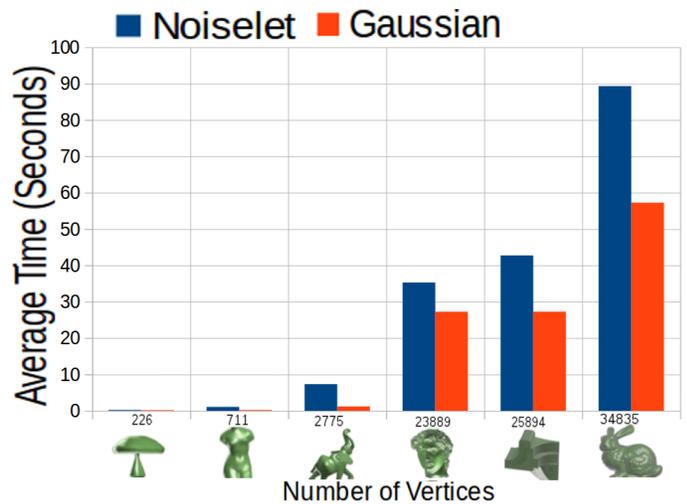


Figure 10: Average of processing times (in seconds ) for the meshes used in our experiment.

by the main geometric compression algorithms found in the literature.

We also measured the average execution time for each mesh and each measurement method. The results are shown in Figure 10. We observed, for both techniques, an increase of the average time as the number of vertices increase. However, the use of Gaussian matrix provided a shorter execution time for all cases. The biggest mesh (Stanford Bunny’s mesh), for example, took 55 seconds, on average, to process, when the Gaussian matrix was used; and 90 seconds, when the Noiselet matrix was employed.

Compared with traditional techniques from the theory of signals, in which reconstruction is done by interpolation, the cost minimization performed in CS techniques is still high. However, the increase of memory capacity and speed of today’s processors tends to minimize this problem.

## VI. CONCLUSION AND FUTURE WORK

Compressive Sensing is a recent paradigm for signal acquisition. Its application is an interesting way to obtain a compressive representation of a signal from fewer measurements than those used in the Shannon-Nyquist theorem. The basics of CS are sparsity and incoherence. In this study we show that CS can be applied to geometric meshes showing quite comparable results with traditional mesh compression techniques.

Algorithm	Connect. Compress Embed comp. rates (bpv)
Deering [35]	11
Topological surgery [36]	6 max. 2 to 6
Valence coder [15]	2.3 on average
Edgebreaker [37]	3.55 max. 2.1 on average
Valence polygonal [21]	1.8 on average
Valence coder [38]	2.1 on average

Table I: Compression Rate in BPV of the main single rate mesh compression algorithms (extracted from [25])

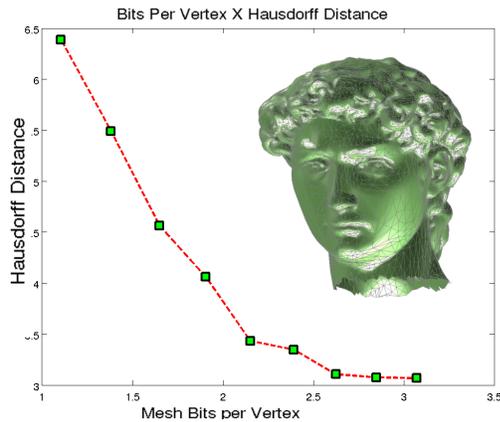


Figure 11: Hausdorff distance depending on the compression rate (BPV) to the David's Head model

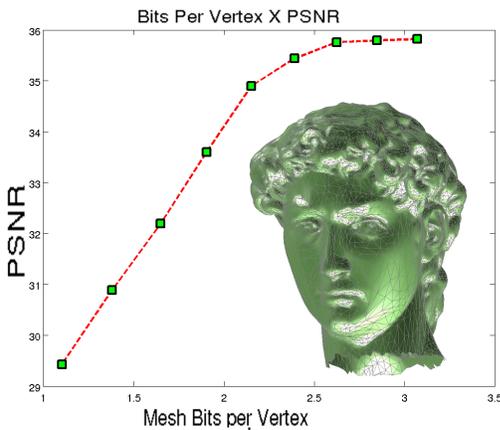


Figure 12: PSNR error depending on the compression rate (BPV) to the David's Head model.

In this work, we used two different settings for our experiments: Gaussian matrix with  $L_1$  norm minimization and Noiselet matrix, using  $L_1$  norm minimization. The results obtained with the application of CS suggest that the use of Gaussian matrix provides a more efficient set-up than that obtained with the use of Noiselet matrices.

The limitations applying CS to meshes are: the high cost of reconstruction, because of the necessity of using optimization methods; and the increasing amount of RAM memory for storing measurement matrices that become larger and larger as the number of vertices increases. Although we used only  $L_1$  norm minimization, there are algorithms that are faster and

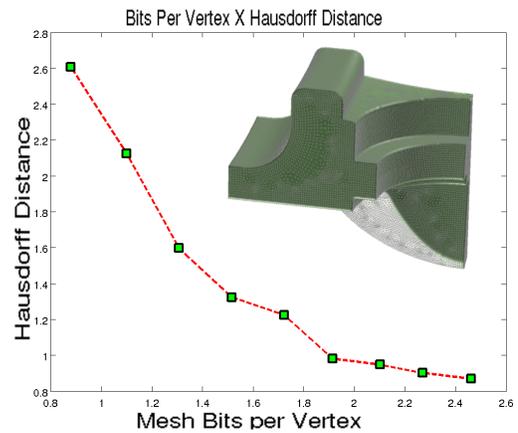


Figure 13: Hausdorff distance depending on the compression rate (BPV) to the Fandisk model

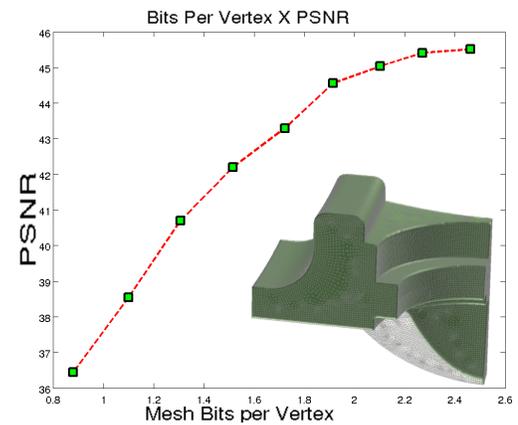


Figure 14: PSNR error depending on the compression rate (BPV) to the Fandisk model.

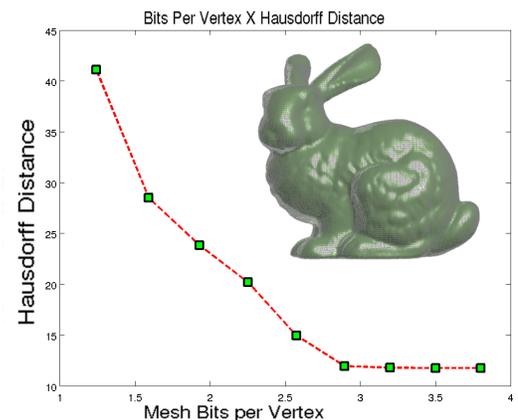


Figure 15: Hausdorff distance depending on the compression rate (BPV) to the Bunny model

provide better minimization results [39].

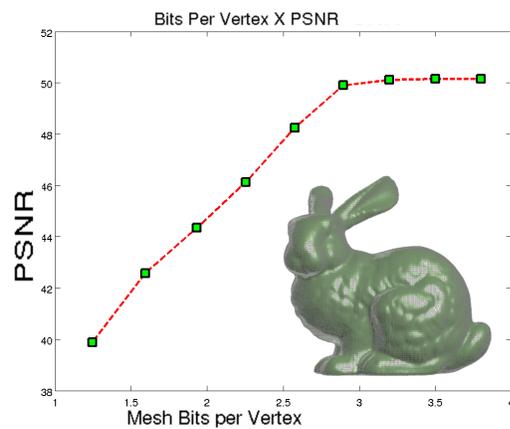


Figure 16: PSNR error depending on the compression rate (BPV) to the Bunny model.

Therefore, we conclude that CS techniques will be useful for geometric meshes because they allow compressive representation of meshes. Also, the reconstructed meshes from CS techniques are reliable and close to the original meshes; and also because the number of bits per vertex used for encoding is comparable to or, some times, better than those provided by classical compression techniques.

For future work, the experiment will be performed with larger meshes, to observe how the technique behaves under those circumstances. Other combinations of measure matrix such Wavelets will be used to compare results. The application of CS in the context of meshes (which mostly represent objects composed of points in  $\mathbb{R}^3$ ) may provide satisfactory results, turning the grid points in a sparse matrix and applying CS. There is a growing interest in the use of three-dimensional models, particularly for entertainment, so the use of compressive representations are relevant. There are a few exploratory studies on the subject [40], thus further study is relevant.

We would also like to study the creation of protocols for compressive acquisition of geometric information, similar to the one described by Duarte *et al.* [11]. With such protocols, it could be possible to acquire geometric information from far fewer sensors than the current process.

## REFERENCES

- [1] E. J. Candès and M. Wakin, "An introduction to compressive sampling [a sensing/sampling paradigm that goes against the common knowledge in data acquisition]," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2007.
- [2] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [3] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, pp. 969–985, 2007.
- [4] A. Schulz, L. Velho, and E. A. B. da Silva, "Uma investigação empírica do desempenho da amostragem compressiva em codificação de imagens," *XXVII SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES*, 2009.
- [5] E. Candès and J. Romberg, "L1-magic : Recovery of sparse signals via convex programming," 2005. [Online]. Available: <http://users.ece.gatech.edu/~justin/l1magic/>
- [6] E. J. Candès, and T. Tao, "Near-optimal signal recovery from random projections and universal encoding strategies?" *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 5406–5445, 2006.
- [7] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, "Distributed compressed sensing of jointly sparse signals," *Conference Record of the Thirty-Ninth Asilomar Conference on Signals Systems and Computers 2005*, vol. vol, pp. 1537–1541, 2005. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1600024>
- [8] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5456165>
- [9] M. Lustig, D. L. Donoho, and J. M. Pauly, "Rapid mr imaging with compressed sensing and randomly under-sampled 3dft trajectories," in *in Proc. 14th*, 2006.
- [10] J. Trzasko and A. Manduca, "Highly undersampled magnetic resonance image reconstruction via homotopic l(0) -minimization." *IEEE Transactions on Medical Imaging*, vol. 28, no. 1, pp. 106–121, 2009. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/19116193>
- [11] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, 2008.
- [12] V. Patel, R. Maleh, A. Gilbert, and R. Chellappa, "Gradient-based image recovery methods from incomplete fourier measurements," *Image Processing, IEEE Transactions on*, vol. 21, no. 1, pp. 94–105, Jan 2012.
- [13] L. Fang, S. Li, R. P. McNabb, Q. Nie, A. N. Kuo, C. A. Toth, J. A. Izatt, and S. Farsiu, "Fast acquisition and reconstruction of optical coherence tomography images via sparse representation." *IEEE Trans Med Imaging*, vol. 32, no. 11, pp. 2034–49, 2013.
- [14] S. Lingala and M. Jacob, "Blind compressive sensing dynamic mri," *Medical Imaging, IEEE Transactions on*, vol. 32, no. 6, pp. 1132–1145, June 2013.
- [15] C. Touma and C. Gotsman, "Triangle mesh compression," in *Proceedings of Graphics Interface*, 1998, pp. 26–34.
- [16] S. Gumhold and W. Strasser, "Real time compression of triangle mesh connectivity," in *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1998, pp. 133–140.
- [17] J. Rossignac, "Edgebreaker: Connectivity compression for triangle meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 47–61, 1999.
- [18] M. Isenburg and J. Snoeyink, "Spirale reversi: Reverse decoding of the edgebreaker encoding," in *12th Canadian Conference on Computational Geometry*, 2000, pp. 247–256.
- [19] D. King and J. Rossignac, "Guaranteed 3.67v bit encoding of planar triangle graphs," in *11TH CANADIAN CONFERENCE ON COMPUTATIONAL GEOMETRY (CCCG 99)*, 1999, pp. 146–149.
- [20] A. Szymczak, D. King, and J. Rossignac, "An edgebreaker-based efficient compression scheme for regular meshes," *Comput. Geom. Theory Appl.*, vol. 20, no. 1-2, pp. 53–68, 2001.
- [21] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive geometry compression," *SIGGRAPH '00 Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 271–278, 2000.
- [22] Z. Karni and C. Gotsman, "3d mesh compression using fixed spectral bases," in *No description on Graphics interface 2001*, ser. GRIN'01. Toronto, Ont., Canada, Canada: Canadian Information Processing Society, 2001, pp. 1–8. [Online]. Available: <http://portal.acm.org/citation.cfm?id=780986.780988>
- [23] 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*, 2011, pp. 62–66.
- [24] P. Lobaz and L. Vása, "Hierarchical laplacian-based compression of triangle meshes," *Graphical Models*, vol. 76, no. 6, pp. 682 – 690, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1524070314000502>
- [25] A. Maglo, G. Lavoué, F. Dupont, and C. Hudelot, "3d mesh compression: Survey, comparisons, and emerging trends," *ACM Comput. Surv.*, vol. 47, no. 3, pp. 44:1–44:41, Feb. 2015. [Online]. Available: <http://doi.acm.org/10.1145/2693443>

- [26] MATLAB, *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010.
- [27] R. Coifman, F. Geshwind, and Y. Meyer, "Noiselets," *Applied and Computational Harmonic Analysis*, vol. 10, pp. 27–44, 2001.
- [28] Y. Gao, M. Wang, R. Ji, X. Wu, and Q. Dai, "3d object retrieval with hausdorff distance learning," *Industrial Electronics, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.
- [29] G. Peyré, "Fourier on meshes," 2008, acesso em Novembro de 2013. [Online]. Available: <https://www.ceremade.dauphine.fr/~peyre>
- [30] D. Salomon, *Data Compression: The Complete Reference*. Springer, 2007, with contributions by Giovanni Motta and David Bryant.
- [31] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. Akar, G. Triantafyllidis, and A. Koz, "Coding algorithms for 3d tv :a survey," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 11, pp. 1606–1621, 2007.
- [32] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive geometry compression," in *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 271–278.
- [33] J. Lee, S. Choe, and S. Lee, "Compression of 3d mesh geometry and vertex attributes for mobile graphics." *JCSE*, vol. 4, no. 3, pp. 207–224, 2010.
- [34] A. Schulz, L. Velho, and E. A. B. Da Silva, "On the empirical rate-distortion performance of compressive sensing," in *Proceedings of the 16th IEEE International Conference on Image Processing*, ser. ICIP'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 3013–3016. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1819298.1819573>
- [35] M. Deering, "Geometry compression," *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, SIGGRAPH '95*, vol. 29, pp. 13–20, 1995.
- [36] G. Taubin and J. Rossignac, "Geometric compression through topological surgery," *ACM Transactions on Graphics (TOG)*, vol. 17, pp. 84–115, 1998.
- [37] J. Rossignac, "Edgebreaker: Connectivity compression for triangle meshes," GVU Center, Georgia Institute of Technology, Tech. Rep., Julho 1998.
- [38] P. Alliez and M. Desbrun, "Progressive compression for lossless transmission of triangle meshes," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 195–202. [Online]. Available: <http://doi.acm.org/10.1145/383259.383281>
- [39] E. Candés, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l-1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, 2008.
- [40] S. M. Yoon and A. Kuijper, "View-based 3d model retrieval using compressive sensing based classification," in *Image and Signal Processing and Analysis (ISPA), 2011 7th International Symposium on*, sept. 2011, pp. 437–442.