3-D Geometric Signal Compression Method Based on Compressed Sensing

Zhuo-Ming Du and Guo-Hua Geng

School of Information Science and Technology, Northwest University, Xi'an, China

Email: duzhuoming423dzm@126.com and ghgeng@nwu.edu.cn

***Abstract*—This paper provides a compression method of three-dimensional meshes based on compressed sensing. First, this method gets the 3-D geometric signal through discrete representing the three-dimensional meshes. Then, we construct a basis using Laplace operator of the three-dimensional meshes. Thus, we get the sparse representation of the 3-D geometric signal based on this basis. Last, we complete compressing the three-dimensional meshes, through random sampling geometry signals based on compressed sensing. In the recovery process, we reconstruct the 3-D geometric signal through optimizing 1-norm of the sparse signal. This method completed the compression of three-dimensional meshes in the sampling process. Experimental results show that the compression ratio of this method is high, the restore effect is good and it is suitable for large-scale data compression.**

Keywords—compressed sensing; geometric signal; random sampling; sparse representation

I. Introduction

Benefiting from the development of scanning techniques, 3D mesh models have recently become a new type of medium after sound, images and video. With the advent of the Web and the increase in demand for geometric models, it is becoming very important to compress 3D mesh models for efficient transmission. The basic content of a 3D mesh model's dataset is the topology, i.e. the connectivity information of the mesh structure, and the geometry, i.e. the coordinates of the model vertices.

All of the works to date on mesh compression [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] have concentrated mostly on efficient coding of the mesh topology, and the secondary coding of the geometry is driven by this. For example, the early mesh compression schemes of Deering [4] and Chow [3] and the later scheme of Taubin and Rossignac [11] ordered the vertices according to the topological information, and then coded them using a simple linear predictor.

Many compression techniques for traditional media, such as images, employ spectral methods to achieve impressive lossy compression ratios, e.g. the popular JPEG method, which relies on the discrete cosine transform. These involve expressing the data as a linear combination of a set of orthogonal basis functions; each basis function is characterized by a "frequency". The underlying assumption is that a relatively good approximation may be obtained using only a small number of low-frequency basis functions. JPEG typically reduces image storage requirements by a factor of 20 relative to the raw RGB data.

Zachi Karni [13] compressed the model using the Spectral method. His method is very much like the JPEG method. But he does not use the discrete cosine transform. He found a new basis. He considered the model as a signal, and projected this signal to the basis he set up. It can complete compressing, only keeping low frequency coefficients, like the JPEG method.

In this paper, we are using the recent method of Compressed Sensing to compress the model. This work uses the basis of Zachi Karni's paper [13].

II. The Theoretical Basis of Our Compression Algorithm

Our algorithm is on the basis of Compressed Sensing Theory, signal representation and spectral transform theory. So we will introduce Compressed Sensing Theory in Section IIA. The three-dimensional meshes are geometric entities. So in Section IIB, we show how to realize their signal representation.

A. Compressed Sensing Background

Compressed Sensing (CS) comprises a collection of methods of representing a signal on the basis of a limited number of measurements and then recovering the signal from these measurements. It is now known that if a signal is measured in terms of independent random projections (i.e., inner products of the signal with random waveforms), then the signal can be reconstructed using these measurements as long as a certain condition that involves the dimension and sparsity of the signal and the number of measurements collected is satisfied [14]-[16]. Algorithms for signal reconstruction in a CS framework are referred to as sparse signal reconstruction (SSR) algorithms. One of the most successful of these algorithms, known as Basis Pursuit (BP), is based on constrained 1-norm minimization.

A real-valued, discrete-time signal represented by a vector x of size N is said to be $K\text{-}sparse$ if it

978-1-61284-881-5/11/$26.00 ©2011 IEEE

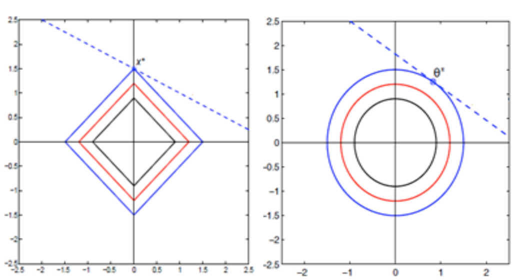
has K nonzero components with $K<<N$. Although most real-world signals do not look sparse under the canonical basis, many natural and man-made signals admit sparse representations with respect to an appropriate basis. For this reason, we can only focus on the class of $K-sparse$ signals. The acquisition of a sparse signal x in CS theory is carried out by obtaining inner products of x with M different waveforms $\{\phi_1 \ \ \phi_2 \ \ .... \ \ \phi_M\}$, namely $y_k = <\phi_k, x>$ for $k=1,2,...,M$. If we let $y = [y_1 \ \ y_2 \ \ ... \ \ y_M]$ and $\Phi = \left[\phi_1^T \ \ \phi_2^T \ \ .... \ \ \phi_M^T\right]^T$, then the data acquisition process in a CS framework can be described as: $y=\Phi x$. Φ is called the measurement matrix. The size of Φ is $M \times N$, typically with $M<<N$. In this way, the signal x is "sensed" by a reduced or "compressed" number of measurements; hence the name of compressive sensing.

With $M<N$, $y=\Phi x$ is an underdetermined system of linear equations, reconstructing the signal x from measurement y is, in general, an ill-posed problem. However, the sparsest solution of $y=\Phi x$ can be obtained by solving the constrained optimization problem:

$$\underset{x}{\text{minimize}} \quad \|x\|_0$$

$$\text{subject to} \quad \Phi x = y$$

where $\|x\|_0$ is the 0-norm of x, defined as $\|x\|_0 = \sum_{i=1}^{N} |x_i|^0$ which, in effect, counts the number of nonzero components in x. Unfortunately optimizing $\|x\|_0$ is a combinatorial optimization problem, where the computational complexity grows exponentially with the signal size, N. Figure 1 shows the solving process to minimize 1-morm and 2-norm.



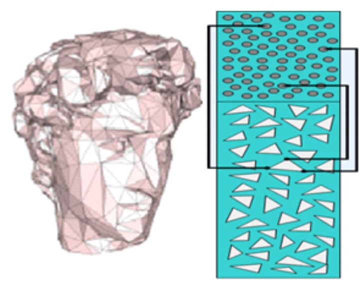
(a) (b)

Figure 1. (a) The solving process of minimize 1-norm; (b) 2-norm. We can see that (a) can get the sparsest solution, while (b) cannot.

B. *Signal Representation of 3D Mesh*

3D mesh can be depicted by a series of triangles, shown in Fig. 2(a). The basic content of a 3D mesh dataset is the topology, i.e. the connectivity information of the mesh structure, and the geometry, i.e. the 3D coordinates of the mesh vertices. The structure of the content of a 3D mesh is shown in Fig. 2(b). Data type of the 3D coordinates of the mesh vertices is float. But the data type of topology is Integer. So, when we compress a 3D mesh, we should focus on its major part, the 3D coordinates.

Let us denote the 3D mesh vertices by a coordinate vector V, which has n rows and three columns, where n is the number of vertices and the three columns correspond to the x, y and z components of the vertex coordinates. Let us denote by x_i (respectively, y_i, z_i) and the x (respectively, y, z) coordinate of a vertex v_i, $i=1,2......,n$. For analysis purposes, let us only consider the x component of V, denoted by X; the treatment of the y and $z$component is similar. We treat the vector X as a discrete 1D signal. Since the mesh is closed, we can view X as a periodic 1D signal defined over uniformly spaced samples along a circle, as illustrated in Figure 3.



(a) (b)

Figure 2. (a) A 3D mesh depicted by a series of triangles; (b) The structure of the content of a 3D mesh. The point is the 3D mesh vertices, which coordinates are (x,y,z). The triangles are the topology of the 3D mesh

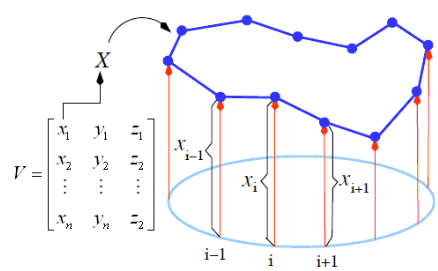


Figure 3. X's coordinate can denote by vector V, which has n rows and three columns

III. Compression Method Based on Compressed Sensing

In this section, we will find a basis from 3D mesh's properties. Under this basis, we complete expressing

3D mesh sparsely. In Section IIIB of this section, we give our compression algorithm.

A. Spectral Transform of 3D Mesh

We treat the 3D mesh as three discrete 1D signals. So we only analyze one of the three signals. The other two are same. Figure 4 shows a plot for the x-coordinates of the 3D mesh (n = 401) Figure 2 (a). The mesh of Figure 2 (a) has 35947 points; we only took part of these points. In order to use compressed sensing to compress this model, we must find a set of bases. Under this set of bases, the model can be represented sparsely.

The mesh M can be described by a pair *(K, V)*, where K describes the connectivity and $V=\{v_1, v_2 \ldots v_n\}$ describes the geometric positions of the vertices in R^3. Each vertex $vi \in V$ has an associated position vector, denoted by $v_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}$; it corresponds to the $i-th$ row of V. Any function defined on the mesh vertices can be seen as a discrete mesh signal. Here we focus on V , the coordinate signal for the mesh. Let us only consider the x component of V, denoted by X; the treatment of the y and z components is similar. Discrete mesh Laplacian operators are linear operators that act on discrete signals defined on the vertices of a mesh. If a mesh M has n vertices, then the mesh Laplacian will be given by an $n \times n$ matrix. Loosely speaking, a mesh Laplacian operator locally takes a weighted average of the differences between the value of a signal at a vertex and its value at the first order or immediate neighbor vertices. We use the following terminology: the neighborhood ring of a vertex i is the set of adjacent vertices $N_i = \{j | (i,j) \in K\}$ and the degree d_i of this vertex is the number of elements in N_i. We assume that the mesh is connected. Let A be the mesh adjacency matrix and $D = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & 0 & \\ & & & d_n \end{bmatrix}$ be the degree matrix. We define L with uniform weights: $L(v_i) = v_i - \frac{1}{d_i} \sum_{j \in N_i} v_j$. So we get the matrix L. And $L = A - \frac{1}{D}$.

L has real eigenvalue and real eigenvector. These eigenvectors consist of a set of bases. Any n-D signal can be represented as the linear combination of these bases. The coefficients of these bases are another kind of expression of the n-D signal. Fortunately, most of these coefficients are zero. So 3D mesh can be represented sparsely under this bases which be consist of the eigenvectors of Laplace operator. Fig. 5 shows a histogram for the x-coordinates of Fig. 2(a) (only has 401 points too), under the new bases. This is the spectral transform of 3D mesh.

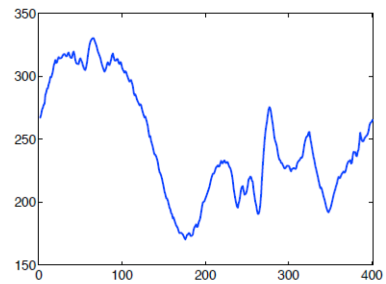


Figure 4. X-coordinates of the seahorse shape

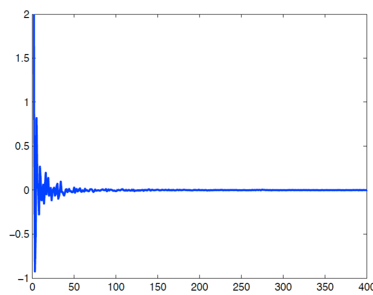


Figure 5. A histogram under the new basis

B. Compression Method Based on Compressed Sensing

From what has been discussed, we know the 3D mesh can be represented as n-D discrete signals. The most important is that this signal is sparse under the basis, which is constituted by eigenvector of Laplace operator. So we can compress this signal using compressed sensing. The compression algorithm of 3D mesh can be expressed:

Step 1: Determine Laplace operator, find the eigenvector of Laplace operator, if the 3D mesh is sparse under the bases be consisted of these eigenvector, go to step2.

Step 2: Project the signal of the 3D mesh to the eigenvector of Laplace operator. The new signals $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ are sparse.

Step 3: Randomly generated matrix $\Psi_{m \times n}$ $m << n$. m must be satisfied with $m \ge C \cdot m \cdot \log(N/K)$. But according to the experience, $m \ge 4k$ is OK, where k is the number of nonzeros of the signal. $\hat{X}$, $\hat{Y}$ and

$\hat{Z}$ sampling respectively using $\Psi_{m\times n}$. $\theta_1 = \Psi_{m\times n} \bullet \hat{X}$, $\theta_2 = \Psi_{m\times n} \bullet \hat{Y}$, $\theta_3 = \Psi_{m\times n} \bullet \hat{Z}$. θ_1 θ_2 and θ_3 are m-D, so the compression is realized.

Step 4: Reconstruct $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ using θ_1 θ_2 and θ_3 .

minimize $\|\theta_1\|_1$

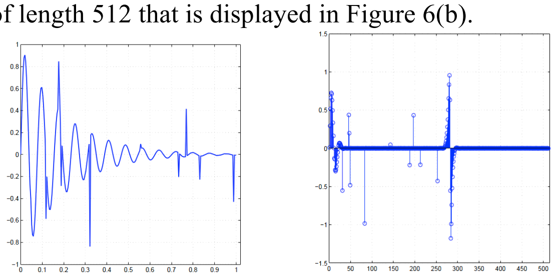
subject to $\theta_1 = \Psi \bullet \hat{X}$. The same method can be used on $\hat{Y}$ and $\hat{Z}$.

Step 5: Reconstruct the signal of the 3D mesh *X* and *Y*. $X = E^{-1} \cdot \hat{X}$, $Y = E^{-1} \cdot \hat{Y}$ and $Z = E^{-1} \cdot \hat{Z}$ E is a matrix which is composed with eigenvector of Laplace operator.

IV. Experimental Results

In this section, we will conduct experiments based on our algorithm. The experiments include two parts, simulation and real experiments. From these results, we can see the simplicity and practicability of the algorithms.

First, for the simulation, we considered the problem of representing a discrete signal *x*, linear combination of atoms of a dictionary. The signal under consideration is of length 256 and is generated by sampling continuous-time signal $e^{-5t}\sin(80.653t)$ at a rate of 255 Hz over time interval [0 1] and then adding a total of 11 spikes of various magnitudes at various sampling instants to it. Figure 6(a) shows the picture. We use $D=\left[I_n C_n^T\right]$ with n = 256 as our dictionary and solve optimization 1-norm as an LP. The solution is a vector θ of length 512 that is displayed in Figure 6(b).

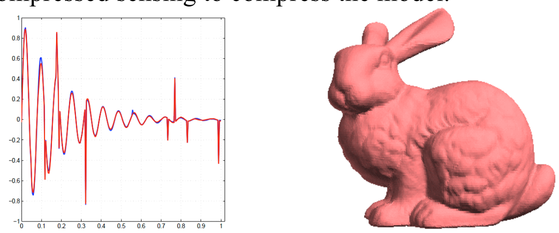


(a) (b)

Figure 6. (a) The original signal of $e^{-5t}\sin(80.653t)$; (b) The sparse expression of this signal

From the figure we see that the solution is quite sparse. By hard-thresholding θ with an $\varepsilon_t = 0.1$, we obtained an even sparser $\hat{\theta}$ with 468 zero entries. If we use $\hat{x} = D\hat{\theta}$ to reconstruct signal *x*, the error is found to be $\left\|x-\hat{x}\right\| / \|x\|_2 = 0.0558$. Signal *x* and its reconstruction $\hat{x}$ are shown in Figure 7(a). We note that among 512 entries of $\hat{\theta}$, there are only 44 nonzero components.

Secondly, we used the same method to 3D mesh. 3D mesh can be expressed as [X Y Z], where *X*, *Y* and *Z* are vectors. But most of their elements are not zero. They are sparse under the basis, which is composed with eigenvector of Laplace operator. *X*, *Y* and *Z* can be expressed as $\hat{X}$, $\hat{Y}$ and $\hat{Z}$. Figure 7(b) shows the original 3D mesh. Table 1 show the vertex statistics of original 3D mesh and sparse 3D mesh under the special basis. From the table, we can see there are almost none zero in the original 3D mesh, and in the sparse 3D mesh, they are almost all zero number. So we can use compressed sensing to compress the model.



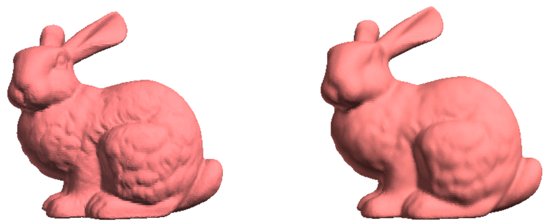
(a) (b)

Figure 7. (a) The reconstruction of signal of $e^{-5t}\sin(80.653t)$; (b) the original 3D mesh

Table 1. Vertex Statistics of Original 3D Mesh and Sparse 3D Mesh Under the Special Basis

	Number of points	Non zero point number	Zero point number
Original 3D mesh	35947	0	35947
	35947	0	35947
	35947	0	35947
Sparse 3D mesh	35947	874	35073
	35947	832	35115
	35947	856	35091

Figure 8 gives the reconstructed model using our algorithm. According to the experience, $m \geq 4k$, k is the number of nonzero of the signal. We let $m = \left[3496 \quad 3228 \quad 3424\right]$. The reconstructed model is Fig. 8(a). From the picture, it is almost identical with the original model. Then we let $m = \left[2887 \quad 2596 \quad 2038\right]$. The reconstructed model is Fig. 8(b). So we do not need to determine *m* according to $M \geq C \cdot M \cdot \log(N/K)$. If we let $m \geq 4k$ k is the number of nonzero of the signal, we can get the original 3D mesh model without any loss.



(a) (b)

Figure 8. (a) The reconstruction of 3D mesh when $m = [3496 \quad 3228 \quad 3424]$; (b) The reconstructed model when $m = [2887 \quad 2596 \quad 2038]$

V. CONCLUSION

In this paper, we show how to compress 3D mesh. Our examples show the effectiveness and efficiency of the method for fairly complex input model. We believe that compressed sensing has a lot more potential in digital geometry processing.

ACKNOWLEDGMENT

We would like to thank Du da-wei for helping us with the implementation. Our work is based on the job of Institute of Visualization Technology, and the School of Information Science and Technology, at Northwest University. This work was supported by the National Natural Science Foundation of China under Grant No.60736008.

REFERENCES

[1] V. Bajaj, V. Pascucci and G. Zhuang, “Single resolution compression of arbitrary triangular meshes with properties,” *Proceedings of the Data Compression Conference*, Snowbird, 1999.

[2] M. Chow, “Geometry compression for real-time graphics,” *Proceedings of Visualization ’97*, IEEE, 1997.

[3] D. Cohen-Or, O. Remez, and D. Levin, “Progressive compression of arbitrary triangular meshes.” *Proceedings of Visualization ‘99*, IEEE, 1999.

[4] M. Deering, “Geometry compression,” *Proceedings of SIGGRAPH ‘95*, ,ACM, pp. 13-20, 1995.

[5] S. Gumhold and W. Strasser, “Real time compression of triangle mesh connectivity,” *Proceedings of SIGGRAPH '98*, ACM, pp. 133-140, 1998.

[6] M. Isenburg and J. Snoeyink, “Mesh collapse compression,” *Proceedings of SIBGRAPHI’99 – 12th Brazilian Symposium on Computer Graphics and Image Processing*, pp. 27-28, 1999.

[7] M. Isenberg and J. Snoeyink, “FaceFixer: Compressing polygon meshes with properties,” *Proceedings of SIGGRAPH*, 2000.

[8] D. King and J. Rossignac, “Guaranteed 3.67v bit encoding of planar triangle graphs,” in *Proceedings of 11th Canadian Conference on Computation Geometry*, pp. 146-149, 1999.

[9] J. Li and C. C. Kuo, “A dual graph approach to 3D triangular mesh compression,” in *Proceedings of the IEEE International Conference on Image Processing*, Chicago, 1998.

[10] J. Rossignac, “Edgebreaker: Connectivity compression for triangle meshes,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, 1999.

[11] G. Taubin and J. Rossignac, “Geometric compression through topological surgery,” *ACM Transactions on Graphics*, vol. 17, no. 2, pp. 84-115, 1998.

[12] C. Touma and C. Gotsman, “Triangle mesh compression,” in *Proceedings of Graphics Interface '98*, 1998, pp. 26-34,.

[13] Z. Karni and C. Gotsman, “Spectral compression of mesh geometry,” in *Proc. of ACM SIGGRAPH*, 2000, pp. 279-286.

[14] E. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489-509, Feb. 2006.

[15] D. L. Donoho, “Compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289-1306, Apr. 2006.

[16] 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, Dec. 2006.