Spectral Compression of Mesh Geometry

Zachi Karni **Craig Gotsman**

Computer Science Dept.
Technion – Israel Institute of Technology
Haifa 32000, Israel

1. Introduction

With the advent of the Web and the increase in demand for 3D content, it is becoming increasingly important to compress 3D mesh data for efficient transmission. The basic content of a 3D mesh is the topology, i.e. the connectivity information of the mesh structure, and the geometry, i.e. the 3D coordinates of the mesh vertices. All of the works to date on mesh compression [1,2,3,4,7,9,10,13,14,17,18] 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 [2] and the later schemes by Taubin and Rossignac [17] order the vertices according to the topological information, and then code them using a simple linear predictor. Similarly, the mesh compression scheme of Touma and Gotsman [18] codes the topology as a traversal of the vertices, and the vertex coordinates are coded by predicting them along this traversal using the so-called "parallelogram rule", which better captures the geometry of the mesh surface. In all cases, the prediction errors are then entropy-coded. Due to the topology coding driving the geometry coding, and not the opposite, the geometry code is not optimal. Ironically, the geometry code is potentially far less compact than the topology code (15 bits/vertex vs. 3 bits/vertex on the average), so more effort should be invested in reducing the geometry code than the topology code, but this seems to have been neglected for the most part by contemporary mesh compression algorithms.

While the existing mesh compression algorithms are advertised as being lossless, from a pure theoretical point of view, they are actually lossy. This follows from quantization of the vertex coordinates to finite precision before the actual coding. Typical 3D mesh geometry is coded to 10-14 bits per coordinate, predictive coding subsequently reducing this to approximately half. At these quantization levels, the decoded mesh is usually visually indistinguishable from the original, justifying the use of the term "lossless". More significant loss may be introduced, and the code size reduced, by performing coarser quantization, but this results in a model with a blocky structure, which is very different from the original. Hence these algorithms are not suitable for lossy compression, due to their non-graceful degradation.

Many compression techniques for traditional media, such as images, employ spectral methods to achieve impressive lossy compression ratios. The popular JPEG method 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 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. The next section shows how to extend classical Fourier theory to the case of 3D meshes. Subsequent sections show how to apply this to 3D mesh coding. Similarly to JPEG, we are able to obtain very significant compression ratios at the expense of a very small loss in mesh quality.

2. Mesh Spectra

Imagine a simple graph consisting of n vertices connected in a cycle. The adjacency matrix A of this graph is the circulant n x n matrix induced by the vector [1 0 1] (where the zero coincides with the diagonal). The so-called Laplacian of A is $L=I-\frac{1}{2}A$, and is the analog of the scond spatial derivative. As Taubin [16] pointed out, the traditional cosine basis functions of the Fourier transform are none other than the eigenvectors of this L. The associated eigenvalues are the squared frequencies. Since the rows of L sum to zero, L is singular and has a vanishing eigenvalue, which corresponds to an eigenvector of constant values. The projection of any real n-dimensional vector on this basis vector is just the DC component (mean) of the vector.

This theory may be extended naturally to more general graph topologies, and, in particular, to 3D mesh structures. If A is the adjacency matrix as defined by the n-vertex mesh topology, i.e.

$$A_{ij} = \begin{cases} 1 & j \text{ and } i \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases}$$

and D is the diagonal matrix such that $D_{ii}=d_i$, where d_i is the degree (valence) of vertex i, then $L=I-DA$ is the mesh Laplacian, and the eigenvectors of L form an orthogonal basis of R^n. The eigenvalues may be considered *frequencies*, and the three projections of each of the coordinate vectors of a 3D mesh geometry vector on the basis functions are the *spectrum* of the geometry. Geometries that are smooth relative to the mesh topology should yield spectra dominated by the low-frequency components. Note that there is a separate spectrum for each of the x,y and z components of the geometry, and they could behave differently.

3. Mesh Partitioning

Computing the spectral basis functions involves computing the eigenvectors of a n x n matrix. Ordinarily, this would require $O(n^3)$ time. However, since the Laplacian matrix is sparse (each row has six non-zero entries on the average), this can be done in $O(n)$ time using multi-resolution methods [12]. Nonetheless, when n is large, the numerical stability of these methods breaks down, due to the eigenvalue set becoming too dense, adjacent eigenvalues becoming too close. Hence it is practically impossible to compute Laplacian eigenvectors for meshes containing more than 1000 vertices, and the mesh must be partitioned into submeshes, each of which is treated separately. This, of course, may result in a degradation in coding quality due to "edge-effects" along submesh boundaries, but has the advantage that local properties of the mesh may be captured better. In order to minimize damage, the partition should be well balanced, i.e. that each submesh contain approximately the same number of vertices, and also the number of edges straddling the different submeshes, the *edgecut*, be minimized. A optimal solution to this problem is NP-Complete [6], and algorithms approximating the optimum are an active branch of graph algorithmic research. An algorithm that performs reasonably well on meshes of up to 100,000 vertices is MeTiS [11], for which an optimized linear-time implementation is available. It seems that METIS gives some preference to minimizing the edge-cut over balancing the partition, which is the preference in our application as well. In particular, if the mesh consists of a number of connected components, MeTiS will prefer to partition into these components, unless the balance is significantly violated.

3. Coefficient Coding

The first step in coding the spectral coefficients is to uniformly quantize them to finite precision. This, of course, introduces some loss into the code. Typical quantization levels are between 10 and 16 bits. The second step is to truncate the coefficient vector. The resulting set of integers is then entropy coded using a Huffman or arithmetic coder [15]. It is possible to optimize the tradeoff between coefficient quantization level and the truncation by taking either a small number of high-precision coefficients, or a large number of low-precision coefficients. In practice, the input parameter to the compression procedure is the desired visual distance d from the original. The mesh is partitioned into k submeshes, such that each submesh contains approximately 500 vertices. The visual distance d is divided by $\sqrt{3k}$, and the number of retained coefficients per coordinate per submesh chosen so that the RMS of the truncated coefficients does not exceed this value. We have defined the visual distance between two meshes with identical topology as the average of the simple RMS geometric distance between corresponding vertices, and the RMS distance between the discrete curvatures at corresponding vertices.

4. Results

We present numerical results for some selct models with a variety of geometric properties. Some models are relatively smooth, but have some fine details. Some are more representative of CAD models, with sharp corners and features. Our compression results are compared with results obtained using commercial compression software of Virtue Ltd. (www.virtue3d.com), which incorporates the compression algorithms of Touma and Gotsman [18], and is widely considered to do a very good job. This software is meant primarily for lossless compression, but can introduce loss by aggressively quantizing the geometry before coding.

To measure the performance of our algorithms, we plot the visual distance between the original and the decoded version of the model vs. the code size. The main result is that in the lossy domain, we are able to achieve codes of approximately half the size of those of Touma and Gotsman for comparable visual distance. In the lossless domain, the code lengths converge to comparable lengths, but we still retain a small advantage.

4. 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, pp. 13-20, ACM, 1995.

[5] M. Desbrun, M. Meyer, P. Schroeder and A. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. Proceedings of SIGGRAPH '99, pp. 317-324, ACM, 1999.

[6] M. Garey and D. Johnson, Computers and intractability: A guide to the theory of NP-completeness, Addison-Wesley, 1978.

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

[8] I. Guskov, W. Sweldens and P. Schroeder. Multiresolution signal processing for meshes. Proceedings of SIGGRAPH '99, pp. 325-334, ACM, 1999.

[9] 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.

[10] 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.

[11] G. Karypis and V. Kumar. MeTiS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. Version 4.0, Univ. of Minnesota, Dept. of Computer Science, 1998. Available at http://www-users.cs.umn.edu/~karypis/metis/metis.html.

[12] L. Kobbelt, S. Campanga, J. Vorsatz and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. Proceedings of SIGGRAPH '98, pp. 105-114, ACM, 1998.

[13] 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.

[14] J. Rossignac. Edgebreaker: Connectivity compression for triangle meshes. IEEE Transactions on Visualization and Computer Graphics, 5(1), 1999.

[15] D. Salomon. Data compression: The complete reference. Springer Verlag, 1998.

[16] G. Taubin. A signal processing approach to fair surface design. Proceedings of SIGGRAPH '95, pp. 351-358, ACM, 1995.

[17] G. Taubin and J. Rossignac. Geometric compression through topological surgery. ACM Transactions on Graphics, 17(2):84-115, 1998.

[18] C. Touma and C. Gotsman. Triangle mesh compression. In Proceedings of Graphics Interface '98, pp. 26-34, 1998.