STATIC 3D TRIANGLE MESH COMPRESSION OVERVIEW

Marcos Avilés and Francisco Morán

Grupo de Tratamiento de Imágenes, Universidad Politécnica de Madrid, Spain[€]
{mar,fmb}@gti.ssr.upm.es

ABSTRACT

3D triangle meshes are extremely used to model discrete surfaces, and almost always represented with two tables: one for geometry and another for connectivity. While the raw size of a triangle mesh is of around 200 bits per vertex, by coding cleverly (and separately) those two distinct kinds of information it is possible to achieve compression ratios of 15:1 or more. Different techniques must be used depending on whether single-rate vs. progressive bitstreams are sought; and, in the latter case, on whether or not hierarchically nested meshes are desirable during reconstruction.

Index Terms — 3D mesh compression, connectivity & geometry coding, progressive meshes, subdivision surfaces.

1. INTRODUCTION

Surfaces embedded in three dimensional (3D) space are still extremely often modeled as polygon meshes for simplicity. In fact, as triangle meshes, since triangles are the basic geometric primitives for standard 3D graphics hardware and for many simulation algorithms. This is why much of the effort in the field of static 3D model compression has been devoted to triangle meshes only, and why we will focus on them in this paper, which is structured as follows. Section 2 sets an upper bound to the raw size of a 3D triangle mesh after introducing the essential difference between the two kinds of information it contains: geometry vs. connectivity. Section 3 gives an outlook on the state-of-the-art techniques and achievable bitrates for the coding of those two kinds of information if a single-rate bitstream is enough. Section 4 explains what the limits are when a progressive bitstream is desired, and the difference between progressive vs. hierarchical, wavelet-based coding of meshes. Finally, Section 5 concludes our presentation and Section 6 gives an extensive list of selected references.

2. RAW SIZE OF A 3D TRIANGLE MESH

It would not be fair to compare the sizes of binary bitstreams or files with those of any of the textual formats used for triangle meshes, e.g., VRML [32]. But what practically all those formats have in common is that they represent a triangular mesh by using two separate tables: one for geometry, in which the three coordinates of each vertex are given; and another for connectivity, listing the indices in the vertex table of the three vertices forming each triangle.

If 32-bit floats are used for vertex coordinates and 32-bit integers for triangle-vertex references, a first, ample estimate of the raw (uncompressed) size of a triangle mesh with V vertices and T triangles is $S = 3 \cdot 32\,(V + T)$ bit. Since typical triangle meshes, and especially large ones, have a vast majority of regular (i.e., valence 6) vertices, their number of triangles is approximately twice that of vertices, and $S \cong 288\,V$ bit, i.e. 288 bits per vertex (bpv). A more reasonable estimate can be obtained by arguing, as King [13], that 32 bits are too many for the vertex indices of most meshes, and that $3\lceil \log_2(V) \rceil T$ bits suffice for connectivity information, even without subjecting it to any sophisticated coding (and almost without quantizing the vertex coordinates: see Section 3.2.1). Therefore:

$$S = 3\,(32\,V + \lceil \log_2(V) \rceil\, T)\text{ bit} \cong 3\,(32 + 2\lceil \log_2(V) \rceil)\text{ bpv.}$$

For a rather large mesh with, say, 50 K vertices (and hence around 100 K triangles), this means 192 bpv, from which half go to the geometry and half to the connectivity.

3. SINGLE-RATE COMPRESSION

Given the clear distinction between the connectivity information and the geometry one, most mesh compression techniques treat them separately. Most early work focused on connectivity coding. The coding order of geometry data is then determined by the way the connectivity is coded. Several methods have been proposed recently to increase the coding efficiency of the vertex information by making the geometry drive the mesh traversal.

3.1. Connectivity coding

The problem of compactly encoding the connectivity has been studied extensively as the theoretical problem of short encoding of planar polygon graphs. Tutte [31] enumerated all the different structures that a connectivity graph can assume, showing that, in the case of arbitrary triangle graphs, the encoding consumes at least $\log_2(256/27) \cong 3.245$ bpv.

[€] This work has been partially supported by the Spanish Administration agency CDTI under project CENIT-VISION 2007-1007 and by the *Comunidad de Madrid* under project S-0505/TIC-0223 (Pro-Multidis).

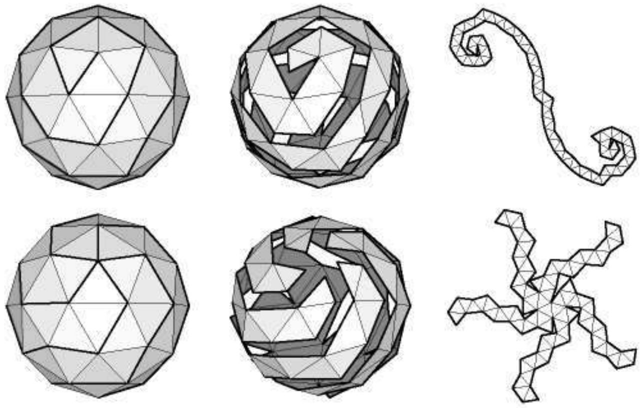

Figure 1: Topological surgery (courtesy of Taubin et al.)

The first attempts were made by Deering [5] and later improved by Chow [4], who accelerated the data transfer from the CPU to a special purpose graphics accelerator that implemented the decoding algorithm. Turán [30] observed that the connectivity of a planar graph can be encoded with a constant number of bpv using two spanning trees: a vertex spanning tree and a triangle spanning tree. Based on this observation, Taubin and Rossignac [26] presented the Topological Surgery (TS) scheme to encode mesh connectivity. The idea is to cut a given mesh along a selected set of cut edges to make a planar polygon, as illustrated by Figure 1. The mesh connectivity is then represented by the structures of cut edges and the polygon, yielding 1 bpv for very regular meshes and 4 bpv on average otherwise.

Subsequently, Rossignac introduced Edgebreaker [23], a simple and efficient traversal technique with an outstanding worst-case bound of 4 bpv. The Mesh Collapse Compression scheme by Isenburg and Snoeyink [11] performs a sequence of edge contractions until a single vertex remains to obtain 1-4 bpv.

Touma and Gotsman [29] pioneered a vertex-based traversal scheme for entropy coding the valence of vertices in triangular meshes with an average of 2 bpv for generic ones, which vanishes to zero for regular grids. It is still considered as one of the best connectivity coders but, more recently, Alliez and Desbrun developed slightly more efficient methods based on a valence-driven approach [2] and on the dual graph concept, introduced in the 3D mesh compression context by Li [18], but employed by them as a means for coding both vertex valences and facet degrees. Furthermore, from the theoretical viewpoint, they proved, together with other co-workers [13], that their connectivity coding techniques are near-optimal, as they achieve Tutte's entropy bound for planar polygonal graphs, under certain reasonable conditions — hence their claim for "near"-optimality.

3.2. Geometry coding

Connectivity reconstruction must frequently be lossless, but state-of-the-art connectivity coders require only a few bits per vertex and their performance is already close to the optimal. On the other hand, geometry data is often given in precise floating point representation and thus eats up quite an important part of the bit budget. Some applications may tolerate some precision loss in order to achieve higher compression rates. Decreasing precision is achieved by means of a *quantization* step, whose resulting values are then compressed by entropy coding after some *prediction* (relying on some smoothness assumptions) is applied. Both steps contribute to the compactness of the final result, but quantization is intrinsically and irreversibly lossy, whereas prediction is a perfectly reversible and lossless transformation of the "signal" to make it fit for a more efficient subsequent entropy coding.

3.2.1. Quantization

Instead of coding the vertex coordinates with the three 32-bit floats mentioned in Section 2, typical mesh geometry coding schemes [5][27][29] uniformly quantize each of them with 8-16 bits. Note that 16-bit integers are enough to resolve 15 μm details in a model of a human body, or 1 mm ones in a model of a big building.

Chow [4] partitions the mesh into several regions and adaptively chooses one quantization resolution per region according to local curvature and triangle sizes. As the human system is more sensitive to normal than to geometric distortion, Sorkine et al. [25] propose to apply the quantization on a transformed coordinate space that concentrates the quantization error at the low-frequency end of the spectrum, thus preserving the fine normal variations over the surface, even after aggressive quantization.

3.2.2. Prediction

Since most edges are short with respect of the model, adjacent vertices are in general close to each other, and the differences between them small. Early work employed simple delta coding [5] or linear prediction [27][29] along the vertex ordering imposed by the coding of the connectivity, yielding bitrates of some 13-18 bpv for 9-12 bits per coordinate [5], or 13 bpv at 8-bit quantization resolution [27]. Despite the effectiveness of these methods, the induced mesh traversal is still not optimal for geometry coding, and that is why more recent techniques [15][17] focus on having the geometry drive the mesh traversal.

4. PROGRESSIVE COMPRESSION

Progressive compression of 3D meshes is desirable for transmission of complex meshes over networks with limited bandwidth or to terminals with limited processing power. The original mesh is transformed into a sequence (or a hierarchy) of refinements applied to a simple, coarse mesh. During decoding, connectivity and geometry are reconstructed incrementally from the bitstream until the mesh is rendered in its full resolution or the transmission is cancelled by the user. Progressive compression thus allows transmission and rendering of different levels of detail (LODs).

4.1. Non-hierarchical refinement and coding

Hoppe [10] first introduced the concept of progressive mesh coding with a new mesh representation called Progressive Mesh (PM), which allows coding a mesh with a total of around 35 bpv, including both connectivity and geometry information. It consists of a base mesh and a sequence of vertex split records, each specifying which vertex and couple of edges incident to it must be split, and the local geometry changes. From such a representation, it is straightforward to extract a LOD of the mesh with any desired number of triangles by simply choosing the adequate prefix of the vertex split sequence, which is streamed after the base mesh has been transmitted.

Based on Hoppe's PMs, and on the TS concept mentioned in Section 3.1, Taubin et al. [28] proposed the Progressive Forest Split (PFS) technique, which was able to reduce the former's bitrates at the expense of reduced granularity: two successive LODs of a PFS set differ by a group of vertex splits, instead of only one. Logically enough, the highest compression ratios are achieved by minimizing the number of LODs but, typically, it is possible to remain slightly below 30 bpv for medium size meshes coded with several LODs. TS and PFS were included in the 3D mesh coding (3DMC) toolset of MPEG-4 version 2 [20].

Pajarola and Rossignac [22] proposed the Compressed Progressive Mesh (CPM) method, by improving on PFS refinements, which also occur thanks to batches of vertex splits. CPM uses the "butterfly" subdivision scheme [6] for predicting the location of new vertices from those of neighboring and already decoded ones, implicitly assuming that the recovered mesh is smooth. CPM achieves bitrates in the vicinity of 25 bpv for medium and large size meshes progressively approximated with 7-15 LODs.

More recently, Alliez and Desbrun presented a novel progressive 3D mesh coding technique [1] with higher compression efficiency and finer granularity than PFS or CPM. This technique yields, thanks to the use of a valence-driven decimation approach leading to near-optimal connectivity coding, and to improvements upon previous geometry compression methods by decorrelating the normal and tangential components of the surface, bitrates of around 20 bpv.

4.2. Remeshing and wavelet-based hierarchical coding

Traditional progressive coders aim to eventually recover the quantized sample locations and original connectivity. For small meshes with carefully laid out connectivity and sample locations, this is very appropriate. The situation is different for highly detailed, densely sampled meshes coming from 3D scanning: since distortion is measured as geometric distance, the sample locations and connectivity can be treated as additional degrees of freedom to improve the rate-distortion performance. As long as the final result has a geometric error similar to that of the original approximation, the actual sample locations and connectivity do not matter.

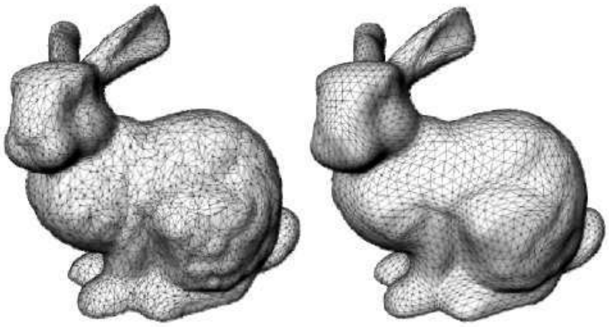

Figure 2: Remeshing of an arbitrary connectivity mesh

Morán [19] and, independently, Khodakovsky [12] proposed slightly different hierarchical compression algorithms based on subdivision surfaces and the wavelet transform. An irregular base mesh, homeomorphic (i.e. topologically equivalent) to the original one, is first obtained by mesh simplification. This base mesh is later refined by recursively subdividing each of its triangles into four and positioning the generated vertices to lie in the original mesh [16]. The arbitrary connectivity original mesh is therefore said to be "remeshed to have semi-regular connectivity": see Figure 2.

The connectivity of the semi-regular mesh can be efficiently encoded, as it only depends on that of the base mesh and the number of subdivisions. The geometry information is that of the base mesh, plus a hierarchical set of wavelet coefficients representing the differences between successive LODs. The distribution of these wavelet coefficients is centered around zero and their magnitude decays at finer levels with a rate related to the smoothness of the original surface. This behavior of the magnitude of wavelet coefficients is the key to hierarchical coding and justifies the choice of a zerotree coder [24]. The results are about four times better (in terms of reconstruction error for a given bitrate) than those of CPM [22], providing even a better performance than Touma's single-rate coder [29].

Guskov et al. [9] proposed another wavelet coder based on the normal mesh representation. During the subdivision step, their algorithm restricts the position of the new vertices to be in the normal direction of the surface. Therefore, only 1D coefficients are needed, which results in a 2-5 dB quality improvement for the same bitrate, compared to the 3D coefficients used by Khodakovsky.

The above techniques only produce signal-to-noise ratio (SNR) scalable bitstreams. More recently, Avilés [3] proposed an alternate wavelet-based coding technique targeted at providing both spatial and SNR scalabilities, and thus enabling the use of the same stream over heterogeneous networks, and with a wide range of terminals. Gioia [7] also carried some remarkable work on view-dependent transmission of wavelet-coded 3D models, whose outcome inspired the `WaveletSubdivisionSurface` node of MPEG-4's AFX toolset [21].

Gu et al. [8] devised a technique for completely regular remeshing of surface meshes using a rectangular grid. Surfaces of arbitrary genus must be cut to be homeomorphic to a disc, then parameterized by minimizing a geometric-stretch measure, and finally represented as a so-called geometry image that stores the geometry, the normals and any attributes required for visualization purposes. Due to its regular structure, geometry images can be compressed using standard 2D image compression techniques.

5. CONCLUSION

This paper gives an overview on the state-of-the-art techniques for the compression of static 3D triangle meshes, which are still extremely used to model discrete surfaces. The uncompressed size of a triangle mesh is in the order of 200 bits per vertex (bpv), but compression ratios of 15:1 or more can be achieved. Different techniques must be used depending on whether single-rate vs. progressive bitstreams are sought; and, in the latter case, on whether or not hierarchically nested meshes are desirable during reconstruction.

Triangle meshes carry two distinct kinds of information, geometry and connectivity, that are best compressed separately, especially if the latter must be reconstructed exactly, and therefore coded losslessly (geometry is always coded in a lossy way due to vertex coordinate quantization). In this case, a total bitrate of around 15-20 bpv is achievable thanks to the best connectivity coders [2][29]. However, if a progressive transmission [1][22] is sought, the total bitrate usually exceeds 20 bpv and the connectivity of the original mesh is only recovered at the end. Moreover, if the original mesh can be remeshed because its connectivity is not essential, wavelet coding techniques based on subdivision surfaces [12][21] offer further compression.

6. REFERENCES

NB: *SG = Proc. ACM SIGGRAPH; EG = Proc. Euro-Graphics.*

[1] P. Alliez and M. Desbrun, "Progressive Compression for Lossless Transmission of Triangle Meshes," *SG*, 195–202, August 2001.

[2] P. Alliez and M. Desbrun, "Valence-Driven Connectivity Encoding of 3D Meshes," *EG*, 480–489 , September 2001.

[3] M. Avilés, F. Morán and N. García, "Progressive Lower Trees of Wavelet Coefficients: Efficient Spatial and SNR Scalable Coding of 3D Models," *Proc. Pacific-Rim Conference on Multimedia*, 61–72, November 2005.

[4] M. Chow, "Optimized Geometry Compression for Real-Time Rendering," *Proc. IEEE Visualization*, 347–354, October 1997.

[5] M. Deering, "Geometry Compression," *SG*, 13–20, July 1995.

[6] N. Dyn, D. Levin and J. A. Gregory, "A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control," *ACM Tr. Graphics*, **9**-2, 160–169, April 1990.

[7] P. Gioia, O. Aubault and C. Bouville, "Real-time Reconstruction of Wavelet Encoded Meshes for View-Dependent Transmission and Visualization," *IEEE Tr. Circuits and Systems for Video Technology*, **14**-7, 937–949, July 2004.

[8] X. Gu, S. Gortler and H. Hoppe, "Geometry Images," *SG*, 355–361, July 2002.

[9] I. Guskov, K. Vidimče, W. Sweldens and P. Schröder, "Normal Meshes," *SG*, 95–102, July 2000.

[10] H. Hoppe: "Progressive Meshes," *SG*, 99–108, August 1996.

[11] M. Isenburg and J. Snoeyink, "Mesh Collapse Compression," *Proc. SIBGRAPI*, 27–28, October 1999.

[12] A. Khodakovsky, P. Schröder and W. Sweldens, "Progressive Geometry Compression," *SG*, 271–278, July 2000.

[13] A. Khodakovsky, P. Alliez, M. Desbrun and P. Schröder, "Near-Optimal Connectivity Encoding of 2-Manifold Polygon Meshes," *Graphical Models*, **64**, 147–168, May 2002.

[14] D. King and J. Rossignac, "Optimal Bit Allocation in 3D Compression," *Computational Geometry: Theory and Applications*, **14**-1-3, 91–118, November 1999.

[15] B. Kronrod and C. Gotsman, "Optimized Compression of Triangle Mesh Geometry Using Prediction Trees," *Proc. 1st International Symposium on 3D Data Processing, Visualization and Transmission*, 602–608, June 2002.

[16] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar and D. Dobkin, "MAPS: Multiresolution Adaptive Parameterization of Surfaces," *SG*, 95–104, July 1998.

[17] H. Lee, P. Alliez and M. Desbrun, "Angle-Analyzer: A Triangle-Quad Mesh Codec," *EG*, 383–392, September 2002.

[18] J. Li and C.-C. J. Kuo, "Progressive Coding of 3-D Graphic Models," *Proc. IEEE*, **86**-6, 1052–1063, June 1998.

[19] F. Morán, "Hierarchical 3D mesh coding with subdivision surfaces," *Proc. Intl. Workshop on Synthetic-Natural Hybrid Coding and 3D Imaging*, 189–192, September 1999.

[20] MPEG (Moving Picture Experts Group, i.e., ISO/IEC JTC1/ SC29/WG11), "ISO/IEC 14496-2/Amd1," a.k.a. "MPEG-4 Part 2, Amendment 1: Visual Extensions," February 2000.

[21] MPEG, "ISO/IEC 14496-16," a.k.a. "MPEG-4 Part 16: Animation Framework eXtension (AFX)," February 2004.

[22] R. Pajarola and J. Rossignac, "Compressed Progressive Meshes," *IEEE Tr. Visualization and Computer Graphics*, **6**-1, 79–93, March 2000.

[23] J. Rossignac, "Edgebreaker: Connectivity Compression for Triangle Meshes," *IEEE Tr. Visualization and Computer Graphics*, **5**-1, 47–61, January 1999.

[24] J. M. Shapiro: "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Tr. Signal Processing*, **41**-12, 3445–3462, December 1993.

[25] O. Sorkine, D. Cohen-Or and S. Toledo, "High-Pass Quantization for Mesh Encoding," *Symposium on Geometry Processing*, 42–51, June 2003.

[26] G. Taubin and J. Rossignac, "Geometric Compression through Topological Surgery," *ACM Tr. Graphics*, **17**-2, 84–115, April 1998.

[27] G. Taubin, W. Horn, F. Lazarus and J. Rossignac, "Geometry Coding and VRML," *Proc. IEEE*, **96**-6, 1228–1243, June 1998.

[28] G. Taubin, A. Guéziec, W. Horn and F. Lazarus, "Progressive Forest Split Compression," *SG*, 123–132, July 1998.

[29] C. Touma and C. Gotsman, "Triangle Mesh Compression," *Proc. Graphics Interface*, 26–34, June 1998.

[30] G. Turán, "On the Succinct Representations of Graphs," *Discrete Applied Mathematics*, **8**-3, 289–294, July 1984.

[31] W. T. Tutte, "A Census of Planar Triangulations," *Canadian Journal of Mathematics*, **14**, 21–38, 1962.

[32] VRML (Virtual Reality Modeling Language, now Web3D) Consortium and ISO/IEC JTC1/SC24, "ISO/IEC 14772," a.k.a. "VRML97," a.k.a. "VRML 2.0," December 1997.