A Brief History of 3D Mesh Compression

Mustafa ORAL[1], Ammar Abbas ELMAS[2]
[1]*Çukurova University Computer Engineering Department, Turkey; moral@cu.edu.tr*
[2]*Çukurova University Computer Engineering Department, Turkey; aelmas@cu.edu.tr*

Abstract

3D graphics are evolving media type used in all aspects of technological areas of today. Increase in demand on 3D graphics pushes technological advancements on 3D scan technology and approximation methods to next level which then results in more complex and highly detailed large 3D raw data. Thus, it is crucial to compress these graphics data efficiently. Over the last two decades, many algorithms have been proposed to compress these raw 3D data especially for compact storage, fast transmission, and efficient processing. Compression methods are branching among themselves. In this paper, 3D compression methods are summarized in a taxonomical fashion. A special attention is paid to the main ideas behind the single-rate compression algorithms and their contribution to 3D mesh compression technology. The advantages and the drawbacks of each algorithm are discussed to pave the road for the future 3D compression researchers.

Keywords: mesh compression, 3D scan, polygonal mesh, triangulation, single-rate

1. INTRODUCTION

Nowadays 3D data can easily be confronted on anywhere in any complexity such as; aerospace models, automotive CAD datasets, architectural walkthrough, virtual environments, computer games, scientific simulations, medical imaging, etc. With the help of evolving 3D scan technology, 3D graphics have gained widespread acceptance. The advancements in modeling algorithms and methods lead us to deliver highly complex 3D models that require a considerable amount of space and bandwidth while transferring and visualizing data especially on a network.

3D models consist of enormous amount of data that need to be represented with proper methods. 3D meshes are by far the most popular polynomial discrete representation method of 3D surfaces. Among the polynomial representation methods, triangulation has been preferred due to their algorithmic simplicity, ease of calculations on GPU side, and displaying efficiency.

Current high-tech graphics cards are partially specialized in rendering this 3D representation method and become available in all parts of our life like smartphones, tablets, personal computers, virtual reality goggles, smart watches etc. Thanks to these graphics cards and various algorithms 3D models can be visualized or edited by special softwares almost on everywhere. On the other side, the large size of 3D mesh data force this active research area to expand on compression in order to satisfy demands on 3D graphics.

2. BACKGROUND & MESH BASICS

Triangular mesh consists of three entities: vertices, edges, and faces. Edges are lines that connecting vertices. Faces are closed surfaces formed by edges. On its most basic form, triangular meshes are represented by *geometry* and their *connectivity* (also called topology or structure) information. Geometry describes point locations on 3D cartesian space for each vertex and may also describes normal vector values for each face. Besides that, connectivity specifies adjacency relationship of mesh elements.

Vertex *valence* (degree) represents number of neighboring edges connected to a vertex. *Regular meshes* are the ones that all faces and vertices have the same size and the same valence, respectively. Most of the research on this topic mainly working on *manifold* meshes which can be defined and distinguished easily on Figure 1.

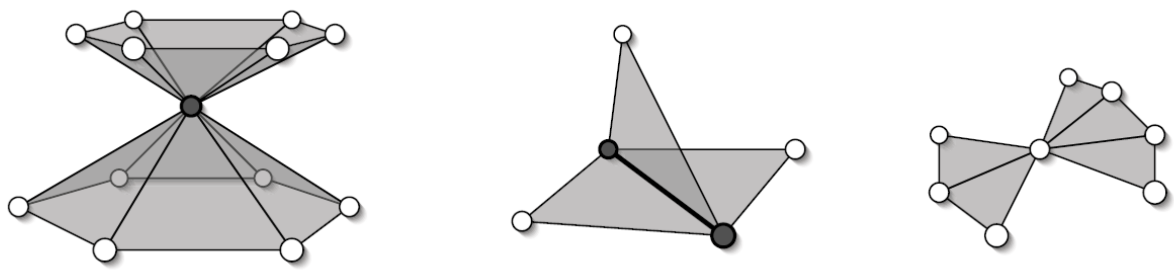


Figure 1 Non-manifold vertex (left), A non-manifold edge (center) has more than two incident faces, (right) example configuration of a non-manifold but can be handled by most of the data structures [1]

3. SINGLE-RATE MESH COMPRESSION

Single-rate (mono resolution) compression algorithms require all the geometry and connectivity information of the model as a whole to encode/decode. Later progressive meshes have demolished this dependency. Most of the mesh compression algorithms treat geometry and connectivity information separately. Early works are mostly focused on connectivity information which is by far the biggest part of 3D raw data. The efficiency of a compression algorithm is usually compared and measured by **b/v** that shows how many bits per vertex are used to encode a 3D mesh. The standard representation for uncompressed polygon meshes uses a list of vertex coordinates to store geometry and a list of vertex indices for each face to store mesh connectivity.

3.1 Geometry Compression

Geometry data of vertex coordinates are often stored in 3-tuple (x, y, z) that is coded in IEEE 32-bit floating point representation. Thus, it dominates quite an important part of the whole 3D data. Also, geometry compression is challenging, because, it deals with floating point numbers rather than integers as in connectivity compression. The 8-bit exponent of 32-bit IEEE floating-point numbers allows positioning of the known universe: from 15 billion light years, down to the sub-atomic particles. That much precision is, obviously, not needed for 3D modelling. Reducing precision by applying quantization can significantly lessen data size without recognizable quality loss. Some applications tolerate a certain amount of precision loss in order to achieve higher compression rates.

Generally, geometry compression begins with quantization of vertex coordinates. Afterwards, rather than encoding point coordinates directly, it uses a prediction scheme to locate next vertex point with the help of already encoded neighbors. There are various quantization methods, including Delta Difference Quantization, Separate Quantization, Global Quantization, and Vector Quantization etc.

3.2 Connectivity Compression

Efficient encoding of the mesh connectivity has been studied extensively. Previous researches on single-rate compression have been mostly dedicated to connectivity coding and many techniques have been proposed and most of them were designed for fully triangulated meshes.

The connectivity information summarizes which mesh elements are connected to each other. Faces are surrounded by its composing edges and all the vertices of its incident edges. The edges have no direction. Two types of mesh connectivity are common in mesh representations. One of them is *edge connectivity* which is list of edges in the mesh and the other is *face connectivity* which is list of faces in the mesh.

4. ALGORITHMS

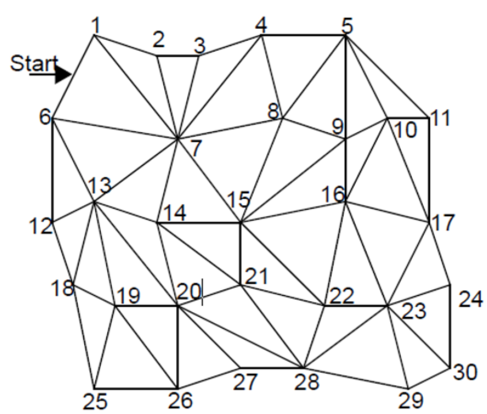
Main approaches are based on triangle strips, spanning trees, triangle traversal, and valence encoding. Pioneering triangle based connectivity driven single-rate mesh compression algorithms according to main approaches are given in the following.

4.1 A triangle strip based encoding algorithm; Geometry Compression [2] (GC)

In 1995 Deering proposed *Geometry Compression* algorithm which then led researchers to work harder on 3D mesh compression field for better compression rates. GC, first, converts triangle mesh data into generalized triangle strip format that can be seen in Figure 2. Triangle strips are sequence of vertices where each new vertex defines a new triangle connected to previous triangle with two previously known vertices. Each triangle is usually adjacent to the previous triangle by using second and third vertices of previous one. Connectivity information is kept on triangle strip form. Triangle strips do not pay off, if we cannot build long enough strips, which is a challenging computational geometry problem.

On the other hand, geometry information is also extensively processed for GC that uses quantization methods for positions, colors, and normals. The quantized data encoded with delta compression followed by a modified Huffman compression. Empirically most geometry is local, so the delta difference between one vertex and the next was expected to be fit in less than 16 bits in significance.

GC is fast and works on board which made it well suited for hardware implementations. GC algorithm reached 8-11 b/v levels for connectivity information which was quite a good start for 3D mesh compression. Since then, various improvements have been made in Geometry Compression algorithm by several researchers [6][7]. Later on, GC algorithm has been integrated in Java 3D.



Generalized Triangle Strip:
R6, O1, O7, O2, O3, M4, M8, O5, O9, O10, M11, M17, M16, M9, O15, O8, O7, M14, O13, M6, O12, M18, M19, M20, M14, O21, O15, O22, O16, O23, O17, O24, M30, M29, M28, M22, O21, M20, M27, O26, M19, O25, O18

Generalized Triangle Mesh:
R6p, O1, O7p, O2, O3, M4, M8p, O5, O9p, O10, M11, M17p, M16p, M-3, O15p, O-5, O6, M14p, O13p, M-9, O12, M18p, M19p, M20p, M-5, O21p, O-7, O22p, O-9, O23, O-10, O-7, M30, M29, M28, M-1, O-2, M-3, M27, O26, M-4, O25, O-5

Legend:
First letter: R = Restart, O = Replace Oldest, M = Replace Mi
Trailing “p” = push into mesh buffer
Number is vertex number, -number is mesh buffer reference where -1 is most recent pushed vertex.

Figure 2 Generalized Triangle Strip and Mesh - Deering 1995 Geometry Compression [2]

4.2 A spanning tree based encoding algorithm; Geometry Compression Through Topological Surgery (TS) [4]

Topological Surgery algorithm encodes a triangular mesh with about 2.5 to 6 b/v thanks to the spanning trees: a vertex and a triangle spanning tree which can be seen on Figure 3. The idea is to cut a given mesh along a selected set of edges to make a planar mesh. The mesh connectivity is then represented by these cuts and planar mesh, producing 1 b/v for almost regular meshes and 4 b/v on average, otherwise. TS algorithm offered an improved and extended way to use a vertex spanning tree to predict the position of each vertex from its ancestors in the tree. Connectivity encoding is lossless. Geometry is predictively encoded. The correction vectors are entropy encoded. Normals, and colors are quantized. Obtaining the optimal spanning tree is an NP-hard combinatorial problem.

Later on, the researchers [9][10] have suggested other data structure models to save spanning trees. TS algorithm is implemented in MPEG4-3D.

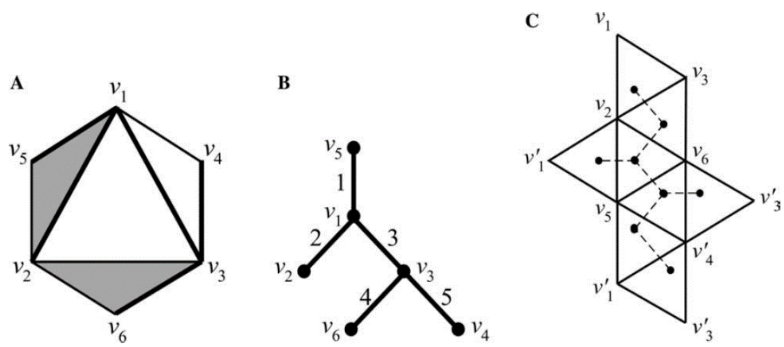


Figure 3 (A) An octahedron mesh, (B) Its vertex spanning tree, and (C) The cut and flattened mesh with its triangle spanning tree shown by dashed lines.[19]

4.3 A triangle traversal based encoding algorithm; Edgebreaker (EB) [5]

Edgebreaker compression stores the connectivity information as CLERS string. It encodes a mesh in a spiraling depth-first spanning-tree traversal order and generates one symbol (either one of C, L, E, R, S) for each triangle. Each symbol represents a relationship between a gate and a vertex on a triangle Figure 4. EB uses its own data structure, corner-table [17], as input. Geometry information is stored in corner table in a predetermined order. EB's encoding algorithm is applied only on connectivity information. Original EB algorithm encodes "C" with one bit, but "L, E, R, S" with three bits which later improved. EB have started another branch on mesh compression which start with 4 b/v for worst case scenario and researcher have enhanced EB algorithm for worst case scenario further with 3,67 b/v and even 3,55 b/v. [11] [12] [13] [14]

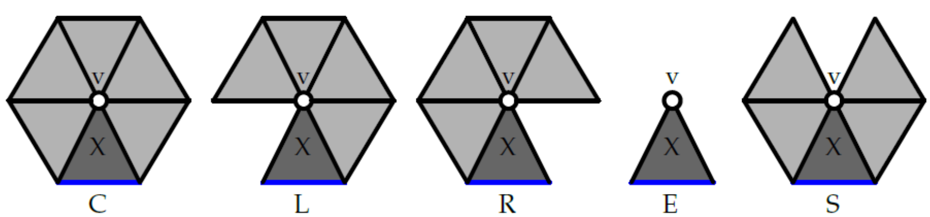


Figure 4 The five configurations of the Edgebreaker algorithm. v is the center vertex and X is the current triangle. The active gate is the blue edge. C: there is a complete triangle fan around v. L: there are missing triangles at the left of the active gate. R: there are missing triangles at the right of the active gate. E: v is only adjacent to X. S: there are missing triangles elsewhere than the left or the right of the active gate. [8]

4.4 A valence based encoding algorithm; Triangle Mesh Compression (TG98) [3]

Triangle Mesh Compression encodes the valence of every vertex along the vertex spanning tree in a depth-first deterministic traversal. The connectivity is encoded by the valence of the inserted vertices, typically accumulated around six connections. Therefore, the generated list of vertex valences can be efficiently compressed by an entropy coder (2.3 b/v). Generally, it stores; the connectivity as a sequence of vertex degrees, geometry as a sequence of vectors which corrects the prediction of a vertex position. TG98 is seen as one of the most efficient connectivity compression method even for today. TG98 not only works on connectivity but also apply prediction algorithm, parallelogram rule, on geometry information.

Later on, it is implemented in Virtue3D. Further improvements and optimality discussions have been made in [15] and [16].

5. CONCLUSION

3D mesh compression mainly focuses on connectivity compression because of the fact that geometry compression does not go further than lossy quantization, prediction and statistical coding methods. Therefore, previously introduced compression methods are evaluated according to compressed connectivity information.

Geometry compression was introduced by Deering 95 in his pioneering work [2]. GC compresses 3D geometry in lossy fashion. A generalized triangular mesh is formed by combining generalized triangle strips with a vertex buffer. GC uses a first-in-first-out (FIFO) vertex buffer to store the indices of up to 16 recently visited vertices. It trims out least significant bits via variable levels of quantization. GC achieves to shrink 3D mesh data down to $1/6^{th}$ - $1/10^{th}$ of original file.

Topological Surgery algorithm relies on ancestors in the tree to predict vertex positions. Thus, it only needs to encode the difference between predicted and actual vertex positions. When vertex coordinates are quantized these corrective vectors have, in average, smaller magnitude than absolute positions and can therefore be encoded with fewer bits.

Edgebreaker algorithm is a finite state machine to compactly describe mesh connectivity that guarantees the 4 b/v worst case scenario. The later researches have decreased this limit to 3.55 b/v [13]. EB can compress the connectivity of the mesh to near optimal rates that is normally around 3 b/v.

Triangle Mesh Compression is accepted as one of the most efficient connectivity compression method even today. Up to now, it is not challenged seriously. Tutte’s entropy [18] that is approximately equal to 3.25 b/v, stands for a theoretical upper bound of the entropy of any arbitrary surface triangular mesh connectivity. A modified version of TG98, that is proposed by Alliez and Desbrun [16], matched to Tutte’s theoretical upper bound entropy. They claimed that valence-based approaches on single rate mesh compression algorithms displays optimal compressions. This achievement, reaching optimum level on single rate mesh compression, led researchers to work on progressive methods.

Table 1 Notes on Pioneering Algorithms

	Method	Connectivity (b/v)
GC - Deering’95 [2]	Triangle Strip	8 – 11
TS - Taubin & Rossignac ’98 [4]	Spanning Tree	6 max – 2.5 to 6
EB - Rossignac ’99 [5]	Triangle Traversal	4 max - 2.1 on average
Touma & Gotsman ’98 [3]	Valence	2.3 on average

REFERENCES

[1] Botsch, M., Pauly, M., Rossl, C., Bischoff, S., & Kobbelt, L. (2006, July). Geometric modeling based on triangle meshes. In ACM SIGGRAPH 2006 Courses (p. 1). ACM.
[2] Deering, M. (1995, September). Geometry compression. In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (pp. 13-20). ACM.
[3] Touma, C., & Gotsman, C. (2000). U.S. Patent No. 6,167,159. Washington, DC: U.S. Patent and Trademark Office.
[4] Taubin, G., & Rossignac, J. (1998). Geometric compression through topological surgery. ACM Transactions on Graphics (TOG), 17(2), 84-115.
[5] Rossignac, J. (1999). Edgebreaker: Connectivity compression for triangle meshes. IEEE transactions on visualization and computer graphics, 5(1), 47-61.
[6] Chow, M. M. (1997, October). Optimized geometry compression for real-time rendering. In Visualization'97., Proceedings (pp. 347-354). IEEE.
[7] Bajaj, C. L., Pascucci, V., & Zhuang, G. (1999). Single resolution compression of arbitrary triangular meshes with properties1This research is supported in part by grants from NSF-CCR-9732306, NSF-KDI-DMS-9873326, DOE-ASCI-BD-485, and NASA-NCC 2-5276.1. Computational Geometry, 14(1-3), 167-186.
[8] Maglo, A., Lavoué, G., Dupont, F., & Hudelot, C. (2015). 3D mesh compression: Survey, comparisons, and emerging trends. ACM Computing Surveys (CSUR), 47(3), 44.
[9] Li, J., & Kuo, C. C. (1998, October). A dual graph approach to 3D triangular mesh compression. In Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on (Vol. 2, pp. 891-894). IEEE.
[10] Diaz-Gutierrez, P., Gopi, M., & Pajarola, R. (2005, September). Hierarchyless Simplification, Stripification and Compression of Triangulated Two-Manifolds. In Computer Graphics Forum (Vol. 24, No. 3, pp. 457-467). Blackwell Publishing, Inc.
[11] Gumhold, S., & Straßer, W. (1998, July). Real time compression of triangle mesh connectivity. In Proceedings of the 25th annual conference on Computer graphics and interactive techniques (pp. 133-140). ACM.
[12] King, D., & Rossignac, J. R. (1999). Guaranteed 3.67 v bit encoding of planar triangle graphs.
[13] Gumhold, S. (2000). New bounds on the encoding of planar triangulations.
[14] Isenburg, M., & Snoeyink, J. (2000, July). Face Fixer: Compressing polygon meshes with properties. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques (pp. 263-270). ACM Press/Addison-Wesley Publishing Co.
[15] Mamou, K., Zaharia, T., & Prêteux, F. (2009). TFAN: A low complexity 3D mesh compression algorithm. Computer Animation and Virtual Worlds, 20(2-3), 343-354.
[16] Alliez, P., & Desbrun, M. (2001, September). Valence-Driven Connectivity Encoding for 3D Meshes. In Computer graphics forum (Vol. 20, No. 3, pp. 480-489). Blackwell Publishers Ltd.
[17] Rossignac, J., Safonova, A., & Szymczak, A. (2003). Edgebreaker on a Corner Table: A simple technique for representing and compressing triangulated surfaces. In Hierarchical and geometrical methods in scientific visualization (pp. 41-50). Springer, Berlin, Heidelberg.
[18] Tutte, W. T. (1962). A census of planar triangulations. Canad. J. Math, 14(1), 21-38.
[19] Peng, J., Kim, C. S., & Kuo, C. C. J. (2005). Technologies for 3D mesh compression: A survey. Journal of Visual Communication and Image Representation, 16(6), 688-733.