

Optimized Prediction for Geometry Compression of Triangle Meshes*

(Abstract)

Dan Chen[†] Yi-Jen Chiang[‡] Nasir Memon[§] Xiaolin Wu[¶]

Recent advances in three-dimensional acquisition, simulation, and design technologies have led to the generation of large polygonal meshes in 3D graphics. The emerging demand for efficiently storing, transmitting, and visualizing such data in networked environments has motivated graphics compression for 3D polygonal models to become a focus of increasing research in the past several years. The most common polygonal models are triangle meshes, consisting of the following two components: *geometry*—the 3D coordinates of the mesh vertices, and *connectivity*—the incidence information specifying the edges and triangle faces connecting the mesh vertices.

Although there has been a significant amount of research done on graphics compression, most previous techniques focus on compressing the *connectivity*, rather than on the *geometry* information. As a result, while connectivity compression already achieves an impressive compression rate of 1.5–4 bits per vertex on an average for triangle meshes [7, 5, 1, 8], progress made in compressing geometry information has not been equally impressive. Typically, a floating-point vertex coordinate is quantized to a 10-bit or 12-bit integer (and hence 30 or 36 bits per vertex for the raw information), and the published techniques that deal with geometry compression (e.g., [2, 3, 7, 8]) generate codes whose size is 40-50% of such raw information. Clearly, geometry compression is the major bottleneck that needs to be worked on if we hope to obtain a significant improvement in the overall compression efficiency.

In this paper, we propose a novel geometry compression technique for 3D triangle meshes. Our approach is based on the *flipping* algorithm first introduced by Touma and Gotsman [8], which traverses the triangles and predicts the position of the next vertex via a *flipping* operation using the *parallelogram rule*. This flipping algorithm [8] has been dominant and widely considered as state of the art for geometry compression, and has been adopted for the MPEG-4 standard for mesh geometry coding [6]. However, like almost all other previous approaches, the original flipping algorithm [8] focuses on connectivity compression and traverses triangles in an order that ignores the geometry of the model, resulting in a geometry compression ratio that is far below optimal as we show in this paper.

The basic idea behind our approach is the observation that for each vertex, there are many different ways to predict its geometry position via the parallelogram-rule flipping operation. We formulate the problem of optimally (traversing triangles and) predicting the vertices via flippings as an optimization problem of a

*Conference version appeared in *Proc. IEEE Data Compression Conference 2005*. It is also available on line at <http://cis.poly.edu/chiang/dcc-paper.ps.gz>. The authors are with the Computer and Information Science Department at Polytechnic University, Brooklyn, NY 11201. Contact author is Yi-Jen Chiang.

[†]dchen@cis.poly.edu. Research supported by NSF Grant CCF-0118915.

[‡]yjc@poly.edu. Research supported in part by NSF Grant CCF-0118915, NSF CAREER Grant CCF-0093373, and NSF ITR Grant CCF-0081964.

[§]memon@poly.edu. Research supported in part by NSF Grant CCF-0118915 and NSF Grant CCF-0208678.

[¶]xwu@poly.edu. Research supported in part by NSF Grant CCF-0208678.

combinatorial nature, namely, the *minimum-spanning-tree* (MST) problem on a graph whose nodes are the mesh vertices and whose edges are all possible flippings with edge cost being the prediction error. However, not all flipping sequences are realizable due to causality constraints. That is, we cannot predict a vertex by a flipping operation unless it is the only unvisited vertex of the parallelogram that defines the underlying flipping operation. Hence we introduce the notion of a *constrained* MST (CMST) which admits a traversal that does not violate causality constraints. We develop a heuristic algorithm to construct an (approximate) CMST, whose cost in practice (as shown in our experiments) is only 17.4% more on an average than the cost of the *unconstrained* MST, which certainly is a severe (unachievable) lower bound on the optimal cost of CMST. Moreover, the flipping operation imposes the causality constraints on the order to *traverse* the constructed CMST; traversing it according to the order in which the tree is constructed is feasible but very expensive to encode. We further develop an encoding algorithm that traverses the constructed tree in another pass, which reduces the encoding cost significantly. The “left-over” triangles that are not yet visited but whose vertices are already spanned are “collected” (for the purpose of connectivity compression) in this same tree-traversal pass. In this way, our compression technique produces a geometry code that is close to optimal (in the sense of what can be achieved by using the flipping operation for prediction), at a slight expense of the cost of the connectivity code. Also, while the encoding step needs more than one pass, decoding needs only one pass and takes linear time.

Our experimental results show the significant advantages of our compression method. In particular, our compression results are up to 33.2% more compact than those of the previous best cover-tree approach [4], and up to 55.4% more compact than those of the widely-used benchmark arbitrary-flipping approach [8].

References

- [1] P. Alliez and M. Desbrun. Valence-driven connectivity encoding for 3D meshes. *Computer Graphics Forum*, 20(3):480–489, 2001. Special Issue for Eurographics ’01.
- [2] M. F. Deering. Geometry compression. In *Proc. ACM SIGGRAPH*, pages 13–20, 1995.
- [3] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *Proc. ACM SIGGRAPH*, pages 279–286, 2000.
- [4] B. Kronrod and C. Gotsman. Optimized compression for triangle mesh geometry using prediction trees. In *Proc. Sympos. on 3D Data Processing, Visualization and Transmission*, pages 602–608, 2002.
- [5] J. Rossignac. Edgebreaker: Connectivity compression for triangle meshes. *IEEE Trans. Visualization Computer Graphics*, 5(1):47–61, 1999.
- [6] MPEG-4 Standard. <http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm>.
- [7] G. Taubin and J. Rossignac. Geometric compression through topological surgery. *ACM Trans. Graphics*, 17(2):84–115, 1998.
- [8] C. Touma and C. Gotsman. Triangle mesh compression. In *Proc. Graphics Interface*, pages 26–34, 1998.