Efficient coding of non-triangular meshes

Boris Kronrod and Craig Cotsman
Computer Science Department
Technion - Israeli Institute Of Technology

1 Contribution

We describe an efficient algorithm for the connectivity coding of general polygon meshes. In contrast to previously published algorithms which are suitable only for triangular meshes, and pay a penalty for treatment of non-triangular facets, this algorithm codes the mesh information in a direct manner. Our treatment of the special case of triangular meshes is equivalent to the Edgebreaker algorithm, albeit our description is somewhat simpler. Using our methods, any regular triangle mesh may be coded in no more than 2 bits/triangle (approximately 4 bits/vertex), any quadrilateral mesh in no more than 3.5 bits/quad (approximately 3.5 bits/vertex), and the most common case of a quad mesh with few triangles in no more than 4 bits/polygon.

2 Introduction

The subject of efficient coding of polyhedral meshes has attracted much interest during recent years, mainly due to the increasing popularity of 3D content on the Web. Without efficient coding methods, it is impossible to transmit this data in reasonable time frames.

Typical 3D content consists of connectivity (topology) information and geometric information. Many algorithms have been designed to code the connectivity information, e.g. the Topological Surgery algorithm of Taubin and Rossignac [9] , the algorithm of Touma and Gotsman [10] Edgebreaker algorithm of Rossignac [8] , its refinement by King and Rossignac [3], and others by Gumhold and Strasser [4], Bajaj et al [11], Deering [2], Chow [1], Isenburg and Snoeyink [5] and Remez et al [7]. The common denominator of these algorithms is that they all deal strictly with inputs which are triangulated meshes, i.e. all mesh faces are triangles. This is crucial for the success of the algorithms, and their extension to handle non-triangulated meshes is not obvious. Hence, in practice, non-triangular meshes are coded by triangulating them, coding the result using one of the methods mentioned above, and storing additional information describing the extra edges introduced during the triangulation stage. These edges are discarded after decoding. Ironically, this means that the code of a non-triangular mesh is larger than that of the triangulated version, instead of being shorter as less connectivity information is present. The only algorithm which extends naturally to non-triangular meshes is that of Li and Kuo [6], using dual-graph methods, but this yields codes whose length could be unbounded.

This abstract proposes a general and direct method for coding the connectivity of a non-triangular mesh with an upper bound on the resulting code length. For the special case of a triangular mesh, it reduces to the Edgebreaker algorithm [8], which bounds the code length from above by 4 bits/vertex.

3 Some Topological Observations

We preface our algorithm by classifying the different interactions possible between an edge cycle in a mesh, and a k-gon face of the mesh, interior to the cycle, whose one base edge e_1, incident on v_0 and v_1, coincides with a cycle edge. For reasons that will become clear later, we call the edge cycle a cut-border. A polygon edge or vertex incident on the cut-border will be called *touching*, otherwise they will be called *free*. If P_k is the number of different types of interaction of a k-gon with a cut-border, then $P_k = 3P_{k-1} - P_{k-2}$. This follows from the fact that for a k-gon, there are three distinct possibilities:

1. The edge e_2 is *touching* . In this case we may chop off the "ear" consisting of the triangle $v_0 - v_1 - v_2$, and remain with a polygon of k-1 edges, such that the original polygon interacts with the original cut-border in precisely the same number of ways that the new polygon interacts with the new (chopped) cut-border. Hence, the contribution of this case to P_k is P_{k-1}.

2. The edge e_2 is *free*, but the vertex v_2 is *touching*. The same argument as in the previous case shows that here too the contribution to P_k is P_{k-1}.

3. The edge e_2 is *free*, and so is the vertex v_2. This is the most interesting case. Here the edge e_3 cannot be *touching*, thus the contribution to P_k is the number of interaction types between the rest of the polygon edges and the cut-border (P_{k-1}) less those for a polygon of degree k-1 with a touching edge (case 1): (P_{k-2}).

By enumeration, it turns out that $P_3 = 5$ and $P_4 = 13$. Since it follows that $P_k < 13 * 3^{k-4}$ for $k > 3$, the number of bits required to code these possibilities is no more than $log_2(13) + (k - 4) * log_2(3)$ for $k > 3$.

4 Encoding

Assume the mesh is manifold and genus-0 with one boundary. If it does not have a boundary (i.e. it is a closed mesh), it is possible to create one by removing one polygon. This boundary will function as the initial cut-border from which polygons will be removed ("cut") one by one. Removing polygons will enlarge the cut-border. Encoding the mesh is a simple matter of traversing the polygons of the mesh via the cut-border and recording the code of the interaction type of the current polygon with the cut-border and the polygon's degree (number of edges). The polygon is removed from the cut-border, the cut-border updated, and the procedure repeated. The procedure terminates when the mesh is reduced to one polygon.

5 Decoding

If for each reconstructed polygon P the following information is given: 1) the cut-border edge touching P's e1. 2) the degree of P. 3) the index of interaction between P and the cut-border. 4) the number of edges in the cut-border before reconstruction of the polygon. 5) the number of edges in the new cut-border(s) generated as a result of the reconstruction of the polygon, the decoding is easy. The first three items are explicit in the code (due to the degree stream, the interaction code stream and the deterministic behavior of the algorithm). Items 4 and 5 may be deduced by the decoder.

Hence the decoder runs in two phases. The first phase computes item 4 and 5 for each decoding step. The second phase reconstructs the connectivity graph using that information.

6 Codebooks and Bounds

Although it is possible to code the meshes using our methods, and then apply entropy coding to take advantage of the relative frequencies of the different symbols, we are able to show that it is possible to use a predefined codebook, and due to the topological properties of manifold meshes, still achieve an upper bound on the total length of the mesh code. This optimization for a triangle mesh is similar to that described in [8], so we will elaborate on quad meshes, and a quad mesh with a minority of triangles. For quad meshes we achieved upper bound 3.5 bits/polygon and for quads with quads with triangles 4 bits/polygon.

References

[1] M. Chow. Optimized geometry compression for real-time rendering. *Proceedings of IEEE Visualization '97*, 1997.

[2] M. Deering. Geometry compression. *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 13–20, 1995.

[3] D.King and J. Rossignac. Guaranteed 3.67v bit encoding of planar triangle graphs. *In Proceedings of 11th Canadian Conference on Computation Geometry*, 1999.

[4] S. Gumhold and W. Strasser. Real time compression of triangle mesh connectivity. In *Proceedings of SIGGRAPH '98.*

[5] M. Isenburg and J. Snoeyink. Mesh collapse compression. In *Proceedings of SIBGRAPHI'99.*

[6] 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.*

[7] D. Cohen-Or O. Remez and D. Levin. Progressive compression of arbitrary triangular meshes. In *Proceedings of Visualization '99.*

[8] J. Rossignac. Edgebreaker: Connectivity compression for triangle meshes. *Transactions on Visualization and Computer Graphics*, 1999.

[9] G. Taubin and J. Rossignac. Geometry compression through topological surgery. *ACM Transactions on Graphics*, 1998.

[10] C. Touma and C. Gotsman. Triangle mesh compression. In *Proceedings of Graphics Interface '98*, 1998.

[11] V. Pascucci V. Bajaj and G. Zhuang. Single resolution compression of arbitrary triangular meshes with properties. *To appear in Computational Geometry: Theory and Applications*, 1999.