Optimized Triangle Mesh Compression Using Prediction Trees

Boris Kronrod Craig Gotsman
Computer Science Department, Technion - Israel Institute of Technology
Haifa 32000, Israel
{kronrod|gotsman}@cs.technion.ac.il

Full version: http://www.cs.technion.ac.il/~gotsman/pubs/kronrod1.zip

1 Introduction

Recently, a wealth of algorithms for the efficient coding of 3D triangle meshes have been published (e.g. [1,2,4,5,6,7,8]). All these focus on achieving the most compact code for the connectivity data, resulting in between 0.7 and 2 bits per triangle on the average. The geometric data, i.e. the vertex coordinates, are then coded in an order induced by the connectivity code, which is probably not optimal. This is a pity, as the geometric portion of the dataset dominates the code.

We propose a way to optimize the geometry code without sacrificing too much in the connectivity code. In our approach we achieve approximately 2 bits/triangle for the connectivity before entropy coding, which is not as good as other published algorithms, but certainly not significantly worse. Our approach is based on the "parallelogram" prediction method for mesh geometry, first introduced by Touma and Gotsman [8]. This method is based on the observation that two adjacent triangles in a typical mesh tend to form a shape similar to a parallelogram. Thus, if a triangle *A* is given, the missing vertex of the adjacent triangle *B* may be predicted quite reliably. If an algorithm uses a parallelogram prediction method, then it should build a traversal structure of triangles covering all vertices. The vertex coordinates are then predicted as the structure is traversed. The cost of this code is then the entropy of the distribution of the vertex prediction errors. Since this entropy is hard to manipulate, the accepted practice is to measure the code effectiveness in the approximation sense, i.e. as the sum of the lengths of the vertex prediction error vectors. The smaller the better.

2 Our Approach

Coding the geometry will require an orderly traversal of the mesh triangles in some order which facilitates accurate prediction of vertex coordinates. This same (tree) structure will also represent a subset of the connectivity information. Hence, we first build a "cover tree" which traverses a subset of the triangles in the mesh, such that all mesh vertices are contained in these triangles at least once.

Unfortunately, there is no easy way to build a cover tree such that all vertices are covered exactly once (this would solve the hard Hamiltonian path problem). Typically a few vertices will be covered more than once. To determine at the decoder which vertices are actually duplicates of vertices decoded before, we identify vertices with the same geometry. The cost of a repeated vertex to the geometry code is not small, but in practice it occurs very infrequently. The cover tree represents only part of the edge information in the mesh. The remainder of the edge information is represented by coding the triangulation of the polygons whose boundaries are the cover tree boundaries.

By using a carefully designed cover tree, of which there are many, we are able to optimize it to achieve better coding of the vertex geometry. Since each vertex's coordinates are predicted from the preceding triangle in the tree, the cost of each tree edge is the prediction error between the two appropriate mesh triangles. The fact that the prediction cost between two adjacent triangles is symmetric implies that the cost of the tree does not depend on which triangle is the root of the tree. Hence the total cost of a mesh cover tree *T*, which we aim to minimize, is the sum of prediction errors on all edges of the tree. Unfortunately, finding this optimal cover tree is a special case of the well-known Minimal Steiner Tree problem for weighted directed graphs, which is NP-Hard [3], so the best that can be hoped for in a polynomial-time algorithm is a cost approximating the optimum.

Also, the cover tree represents only part of the connectivity of the mesh. Thus more information is required to complete the rest of the connectivity information. The boundaries of the cover tree form a set of triangulated polygons. The connectivity of a triangulated polygon may be represented with two bits per triangle, using a binary spanning tree on the triangles.

We cast the problem of finding an optimal cover tree as a graph-theoretic problem. We build a graph G=<V,E>, such that the node set V is the set of all mesh triangles

Proceedings of the Eighth Pacific Conference on Computer Graphics and Applications (PG'00)
0-7695-0868-5/00 $10.00 © 2000 IEEE

and vertices, and E contains a bi-directional edge between two mesh triangles iff they are adjacent. The weight of this edge is the symmetric prediction error of the two triangles. E also contains a directed edge from a triangle to all three of its vertices. The weight of this edge is zero. G is very similar to the dual of the triangle mesh. The objective is to construct a tree on G which covers all nodes of G which are mesh vertices, and has minimal weight. This tree will cover all mesh vertices via a subset of the mesh triangles.

We approximate the optimum with a greedy algorithm using the simple approximation heuristic for the undirected Minimal Steiner Tree problem: Starting from an arbitrary mesh vertex, the tree is extended by the vertex which is closest to the tree (using a path in the graph). This achieves an approximation of the optimum by a factor of two. The complexity of this (encoding) algorithm is $O(n^2)$, but has been found to be significantly less in the typical case. The complexity of the decoding algorithm is linear.

3 Experimental Results

We have implemented the algorithms outlined here, and compared the results with those obtained by the algorithm of Touma and Gotsman [8], which is widely considered to be state-of-the-art. Additionally, we have compared the results to a lower bound computed by considering, for each triangle, the adjacent triangle which gives the best possible prediction. The models tested were first simplified relative to the originals, and the polygon count reduced by 10%. This was done in order to reduce the regularity of the models, which might artificially bias towards exact parallelogram structures in the mesh.

Our results, summarized in Table 1, show that our technique reduces the mean prediction error by up to 15% relative to the algorithm of Touma and Gotsman, but the entropy of the prediction error distribution is only 5% less. This is because the optimization procedure operates directly on the approximation error, and not on the entropy, and the connection between the two is not straightforward. Even so, the approximation errors we achieve are still quite far from the lower bound, so there is probably still some room for improvement.

4 References

[1] V. Bajaj, V. Pascucci and G. Zhuang. Single resolution compression of arbitrary triangular meshes with properties. To appear in *Computational Geometry: Theory and Applications 1999*, also in *Proceedings of Data Compression Conference* , Snowbird, Utah, 1999.

[2] M. Deering, Geometry compression, *Proceedings of Siggraph '95*, ACM, 1995.

[3] M. Garey and D. Johnson, *Computers and Intractability : A Guide to the Theory of NP-Completeness*, pp. 208-209, 1978.

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

[5] M. Isenberg and J. Snoeyink. Face Fixer: Compressing polygon meshes with properties. *Proceedings of SIGGRAPH 2000*.

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

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

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

Table 1: Experimental results of geometry coding (only). TG – Touma and Gotsman algorithm. KG – Our algorithm. All model geometries were quantized to 10 bits/coordinate before coding.

Model	# verts	RMS Prediction Error				Entropy (bits/coord)		
		Bound	TG	KG	%	TG	KG	%
Thorax	1,366	25.2	89.1	70.2	78	7.6	7.0	92
Triceratops	2,549	4.7	68.9	56.7	82	6.6	6.6	100
Face	11,277	7.9	24.7	20.9	85	6.0	5.6	93
Horse	17,866	5.2	19.9	16.7	84	5.7	5.6	98

0-7695-0868-5/00 $10.00