Guaranteed 3.67V bit encoding of planar triangle graphs

Davis King
GVU Center
College of Computing
Georgia Institute of Technology
404/894-0671
kingd@cc.gatech.edu

Jarek Rossignac
GVU Center
College of Computing
Georgia Institute of Technology
404/894-0671
jarek@cc.gatech.edu

ABSTRACT

We present a new representation that is guaranteed to encode any planar triangle graph of V vertices in less than 3.67V bits. Our code improves on all prior solutions to this well studied problem and lies within 13% of the theoretical lower limit of the worst case guaranteed bound. It is based on a new encoding of the CLERS string produced by Rossignac's Edgebreaker compression [Rossignac99]. The elegance and simplicity of this technique makes it suitable for a variety of 2D and 3D triangle mesh compression applications. Simple and fast compression/decompression algorithms with linear time and space complexity are available.

Keywords: *3D representations, triangle meshes, planar graph encoding, geometry compression.*

1. INTRODUCTION

Many 3D models used in engineering, scientific, medical, geographical, and visualization applications are represented by an irregular mesh of bounding triangles. The simplest representation of such a mesh stores *the geometry* (a table of the coordinates of its V vertices) and *the connectivity* (a table of triangles, each represented by three vertex indices of at least $\log_2(V)$ bits each).

The problem of compactly encoding the connectivity has been studied extensively as the theoretical problem of short encoding of planar triangle graphs [Tutte62, Tutte73, Itai&Rodeh82, Turan84, Naor90, Kao&Teng94, Keeler&Westbrook95, Munro&Raman97, Chuang&others98, He&Kao99] and as a practical problem of compressing the incidence table of triangle meshes for 2D and 3D models [Deering95, Chow97, Denny97, Hoppe98, Gumhold98, Li98, Taubin98, Touma98, Rossignac99, Rossignac&Szymczak99].

Because, for large V, the connectivity of triangle meshes dominates storage, it is important to develop compact representations for it. For meshes that are homeomorphic to a sphere, the connectivity information may be represented by a planar triangle graph. Thus, a short encoding of an unlabeled planar graph may be used for compressing such triangle meshes, given a convention for canonical vertex labeling.

Furthermore, in many applications, it is possible to compress vertex coordinates down to 4 or 5 bits each by using vertex quantization, geometric predictors, and variable length encoding of corrective vectors [Deering95, Chow97, Hoppe98, Taubin&Rossignac98].

Most techniques for encoding planar graphs may be extended to handle more general meshes with handles, boundaries, and other non-manifold singularities (see for example discussions in [Rossignac&Cardoze99, Rossignac&Szymczak99]).

While theoretical investigations are focused on lowering the worst case storage cost for any planar triangle graph, practical triangle mesh compression approaches are focused on lowering the expected storage cost for 3D models commonly found in CAD, medical, GIS, animation, and graphic applications.

The work reported here makes a theoretical contribution by introducing a new representation that is guaranteed to encode any planar triangle graph of V vertices in less than 3.67V bits.

Our new code improves on all prior solutions to this well studied problem, including Keeler and Westbrook's bound of 4.6V bits [Keeler&Westbrook95] and Rossignac's recently published bound of 4.0V bits [Rossignac99]. It lies within 13% of the theoretical lower limit of 3.24 bits worst case guaranteed bound, established by Tutte [Tutte62].

Our approach is based on a new encoding of the CLERS string, produced by Rossignac's Edgebreaker compression algorithm [Rossignac99]. A faster, Wrap&Zip, decompression algorithm was recently developed by Rossignac and Szymczak [Rossignac&Szymczak99]. Both algorithms exhibit linear time and space complexity.

We briefly explain the CLERS string and outline both the Egebreaker and the Wrap&Zip algorithms in the next section. Then we describe our new encoding of the CLERS string and justify our claims of 3.67V bits. Finally, we discuss extensions to the connectivity graphs of meshes that are non-homeomorphic to a sphere.

2. EDGEBREAKER

Consider a triangle mesh of T triangles and V vertices that is homeomorphic to a sphere. T equals 2V–4. The triangle/vertex incidence relation may be encoded as a planar triangle graph.

Edgebreaker [Rossignac99] visits the triangles and vertices of the entire mesh in depth-first order by applying the transformations shown in Figure 1. The left column shows the precondition for each one of the 5 Edgebreaker operations, which visit and label one triangle at a time. The right column shows the result of each operation and the label associated with the most recently visited triangle. The CLERS string is the concatenation of the C, L, E, R, or S labels identifying each operation.

We use the following color codes. The current triangle is green. Previously processed triangles are red. References to yellow triangles are pushed on the stack and will be used to identify the current triangle after each E operation. White triangles have not been processed.

Blue circles identify labeled vertices, and an open circle identifies a vertex which must not have been previously labeled. We use incremental labels, which define the order in which vertex coordinates and other attributes are encoded.

The resulting CLERS sequence suffices to capture the connectivity of the mesh (i.e. of the associated planar triangle graph).

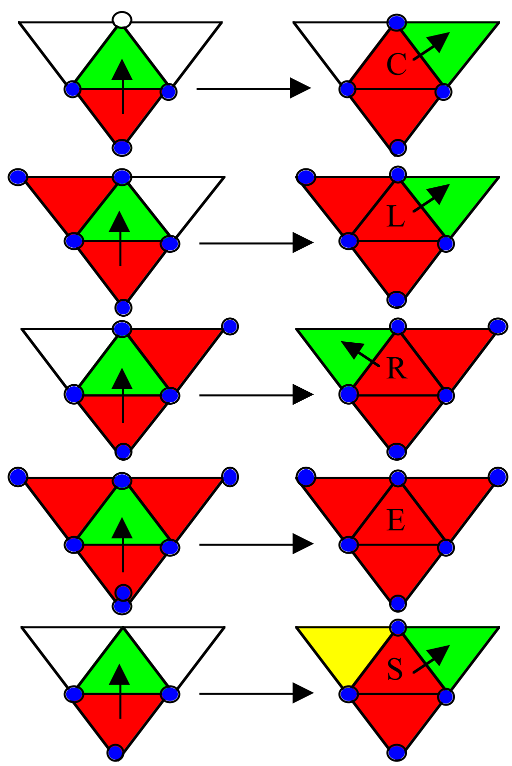

Figure 1: Edgebreaker compression.

A simple encoding of the labels guarantees 4 bits per vertex. It is based on the observation that exactly 50% of the symbols are C operations, because only C operations reach new vertices and because there are two times as many triangles as vertices. Therefore, it suffices to encode each C with a single bit code (for example 0) and each one of the other four operations with a 3-bit code starting with a 1. Such a code is guaranteed to take 2.0T bits, or equivalently 4.0V bits.

More elaborate codes, proposed in [Rossignac99, Rossignac&Szymczak99], further reduce the expected storage cost. For example, the fact that CL and CE combinations are impossible leads to a simple code with an expected storage cost to 3.2V bits. For very large meshes, an entropy code reduces the expected storage cost to less than 2.0V bits [Rossignac&Szymczak99]. However, these improved codes for the expected mesh may require 4V or more bits in the worst situation. Thus these advanced codes do not improve the original Edgebreaker's 4.0V bits worst case bound.

Wrap&Zip decompression [Rossignac&Szymczak99] decodes and uses the labels of the CLERS string to decide where to append each new triangle to a previously reconstructed one. The result is a simply connected topological polygon that corresponds to a triangle-spanning tree of the original mesh. To correctly glue the corresponding pairs of its bounding edges, Wrap&Zip uses the labels to orient the free edges that bound the polygon counter-clockwise for L, R, and E, and clockwise for C triangles, as shown in Figure 2.

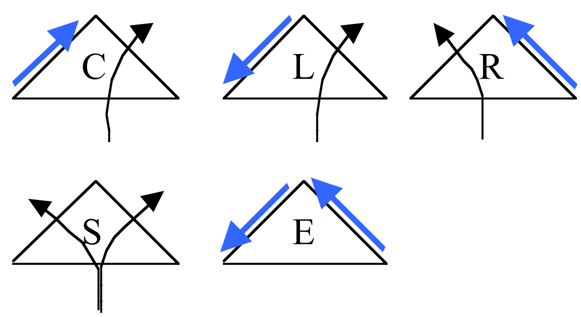

Figure 2: Wrap&Zip free edge orientation.

A recursive procedure restores the complete incidence information by gluing pairs of adjacent edges whose orientations point towards their common vertex. Vertices are decoded in the order in which they are first encountered.

2. OUR NEW CODES

We achieve a guaranteed encoding of 1 and 5/6 bits per triangle by using a 2-bit prefix for the entire CLERS string to select between three alternative codes, of which at least one takes 1 and 5/6 bits per triangle or less.

The three codes, 1, 2, and 3 are presented in Table 1. A C will be denoted C_A when it immediately follows another C and C_N otherwise. We use a similar notation for S and R. L and E can never follow a C.

State:	Code 1:	Code 2:	Code 3:
C_A	0	0	0
S_A	10	10	10
R_A	11	11	11
C_N	0	00	00
S_N	100	111	010
R_N	101	10	011
L	110	110	10
E	111	01	11
Total cost	$2T–\|S_A\|–\|R_A\|$	$2T–\|R_N\|–\|E\|$	$2T–\|L\|–\|E\|$

Table 1: Three codes

Note that during decompression, the previous label is known and hence indicates whether one should use codes for labels that follow a C or codes for labels that do not follow a C. Note that in each case, and for each one of our three coding schemes, these codes are exclusive.

The total cost of these operations in bits (bottom row) is expressed in terms of T and in terms of the number of incidences |X| of labels of type X in the CLERS string. We first justify these total cost formulae and then prove that at least one of them must not exceed 11T/6 bits.

There are exactly 2+T/2 vertices. The first two vertices bound the initial starting edge and need not be explicitly encoded. Therefore, there is exactly T/2 labels of type C and thus |L|+|E|+|R|+|S| = T/2. Consequently, there are always T/2 labels that follow a C and T/2–1 labels that do not. Consequently, $|C_N|+|C_A|=T/2$ and $|S_A|+R_A|+|C_A|=T/2$. From these two equations, we obtain $|C_N|=|S_A|+R_A|$.

By adding the cost of all labels, the total cost in bits for code 1 is $|C_A|+2|S_A|+2|R_A|+|C_N|+3|S_N|+3|R_N|+3|L|+3|E|$. By rearranging the terms we obtain $(|C_A|+|C_N|)+(2|S_A|+|S_A|)+(2|R_A|+|R_A|)$ $+3|S_N|+3|R_N|+3|L|+3|E|–|S_A|–|R_A|$ and by identifying groups C+3(|SA|+|RA|+|SN|+|RN|+|L|+|E|)–(|SA|+|RA|). Given that |C|=T/2, that |S|+|R|+|L|+|E|=T/2, and that $|S_A|+R_A|=|C_N|$ we obtain $2T–(|S_A|+R_A|)$.

The total cost of code 2 is $|C_A|+2|S_A|+2|R_A|+|2C_N|+3|S_N|+2|R_N|$ $+3|L|+2|E|$. We rearrange and group the terms to obtain $(|C_A|+|C_N|) + |C_N|–$ (|SA|+|RA) + 3(|SA| + |RA| + |SN| + |RN| + |L| + |E|) - (|RN|+|E|). The $|C_N|–$(|SA|+|RA|) cancel outs, and, given that |C|=T/2 and |S|+|R|+|L|+|E|=T/2, we obtain 2T-(|RN|+|E|).

The cost of code 3 is $|C_A|+2|S_A|+2|R_A|+|2C_N|+3|S_N|+3|R_N|$ $+2|L|+2|E|$. Rearranging terms, it is $(|C_A|+|C_N|)+|C_N|$-(|SA|+|RA) +3(|SA|+|RA|+|SN|+|RN|+|L|+|E|)-(|L|+|E|). We obtain 2T-(|L|+|E|).

The best alternative among the three codes has cost $\min(2T–|S_A|–|R_A|,\ 2T–|R_N|–|E|,\ 2T–|L|–|E|)$, which may be simplified to $2T\text{-}\max(|S_A|+|R_A|,\ |R_N|+|E|,\ |L|+|E|)$. Clearly, $3\max(|S_A|+|R_A|,\ |R_N|+|E|,\ |L|+|E|)$ is no less than the sum $|S_A|+|R_A|+|R_N|+|E|+|L|+|E|$. Since |E|=|S|+1, this sum equals $|S_A|+|R_A|+|R_N|+|E|+|L|+|S_A|+|S_N|$. Applying the fact that |S|+|R|+|L|+|E|=T/2, we obtain $\max(|S_A|+|R_A|,\ |R_N|+|E|\ ,|L|+|E|)$ >= 1/3 $(T/2 + |S_A|+1) > T/6$.

The total worst-case cost, therefore, is $2T\text{-}\max(|S_A|+|R_A|,$ $|R_N|+|E|,\ |L|+|E|) < 2T – T/6 - |S_A|/3 < 11T/6$, or, less than approximately 1.83 T. For meshes homeomorphic to a sphere, T=2V-4, so the cost is bounded below 3.67 bits per vertex. Since $|S_A|$ may take on any value between 0 and T/4, the $|S_A|/3$ term in the formula may reduce the expected cost but does not improve the worst-case guarantee.

2.1.1 Meshes with Boundary

A mesh with boundary may be encoded by adding a dummy vertex and connecting the dummy vertex to each boundary edge with an additional triangle. If the encoding begins from the dummy vertex, each such triangle will be a C, and the sequence of initial C's may be encoded in O(log(B)) bits, where B is the number of boundary edges. The resulting total cost is 11/6 (T+B) - B + O(log(B)), or 11/6 T + 5/6 B + O(log(B)).

2.1.2 Meshes with Handles and Holes

Handles and holes may be represented by modifying each S to indicate whether the boundary vertex reached is on the boundary of the mesh itself, on the border of a hole, or on a loop that cuts a handle. Rossignac [Rossignac] proposes to store the bits distinguishing among those three cases in a separate table, which may be entropy coded to take advantage of the low frequency of holes and handles in most meshes. The table must also identify which handle has been reached and record the number of edges in the border of each hole, at cost O(log(handles)) and O(holes*log(B)) respectively.

In a mesh with handles, the above triangle frequencies change somewhat. The total number of triangles T is 2V-4+4H, the total number of C's is therefore approximately T/2 – 2H, and the number of non-C's is T/2+2H. The cost of the codes, therefore, becomes $2T+4H\text{-}\max(|S_A|+|R_A|,|R_N|+|E|,|L|+|E|)$. Since the additional triangles are non-C's, however, the maximum becomes at least T/6+H/3, for a cost no worse than 3.67 (V+2H)+0.83B, plus the cost of encoding the lookup table.

3. ACKNOWLEDGMENTS

This work was supported by NSF Grant 9721358.

4. REFERENCES

[1] M. Chow, Optimized Geometry Compression for Real-time Rendering, Proc. IEEE Visualization'97, pp. 347-354, Phoenix, AZ, October 19-24, 1997.

[2] R. C. Chuang, A. Garg, X. He, M. Y. Kao, and H. I. Lu. Compact encodings of planar graphs via canonical

orderings and multiple parentheses. Extended abstract in Proceedings of the 25th International Colloquium on Automata, Languages, and Programming, pages 118-129, 1998.

[3] M. Deering, Geometry Compression, Computer Graphics, Proc. ACM Siggraph'95, 13-20, August 1995.

[4] M. Denny and C. Sohler. Encoding a triangulation as a permutation of its point set. In Proceedings of the Ninth Canadian Conference on Computational Geometry, pp. 39-43, Ontario, August 11-14, 1997.

[5] S. Gumhold, and W. Strasser, "Real Time Compression of Triangle Mesh Connectivity." Proc. ACM Siggraph 98, pp. 133-140, July 1998.

[6] X. He, M. Y. Kao, and H. I. Lu, Linear-time succinct encodings of planar graphs via canonical orderings. SIAM Journal on Discrete Mathematics, (1999), to appear.

[7] H. Hoppe, "Progressive Meshes," Proceedings ACM SIGGRAPH'96, pp. 99-108, August 1996.

[8] A. Itai and M. Rodeh, Representations of Graphs, Acta Informatica, No. 17, pp. 215-219, 1982.

[9] M. Y. Kao, N. Occhiogrosso, and S. H. Teng, Simple and efficient compression schemes for dense and complement graphs, Journal of Combinatorial Optimization, (1999), to appear.

[10] K. Keeler, and J. Westbrook, "Short Encodings of Planar Graphs and Maps," Discrete Applied Mathematics, No. 58, pp. 239-252, 1995.

[11] J. Li, J. Li and C.C Kuo, "Progressive Coding of 3D Graphic Models," IEEE Proc. Multimedia Computing and Systems, pp. 1052-1063. June 1998.

[12] J. I. Munro and V. Raman, Succinct representation of balanced parentheses, static trees and planar graphs, in Proceedings of the 38th Annual IEEE Symposium on the Foundations of Computer Scienece, 1997, pp. 118-126.

[13] M. Naor. Succinct representation of general unlabeled graphs. Discrete Applied Mathematics, vol. 29, pp. 303-307. North Holland, 1990.

[14] J. Rossignac and D. Cardoze, Matchmaker: Manifold Breps for non-manifold R-sets, to appear, Proceedings of the ACM Symposium on Solid Modeling, 1999.

[15] J. Rossignac, Edgebreaker: Connectivity compression for triangle meshes, IEEE Transactions on Visualization and Computer Graphics, Vol. 5, No. 1, January - March 1999.

[16] J. Rossignac, and A. Szymczak, Wrap&zip: Linear decoding of planar triangle graphs, GVU Center, Georgia Institute of Technology, Tech Report: GIT-GVU-99-08, January 1999. http://www.cc.gatech.edu/~jarek/abstracts.html

[17] G. Taubin, A. Gueziec, W. Horn, F. Lazarus, Progressive Forest Split Compression, Proc. ACM Siggraph 98, pp. 123-132, July 1998.

[18] G. Taubin, and J. Rossignac, Geometric Compression through Topological Surgery, ACM Transactions on Graphics, Volume 17, Number 2, pp. 84-115, April 1998.

[19] C. Touma and C. Gotsman, "Triangle Mesh Compression," Proceedings Graphics Interface 98, pp. 26-34, 1998.

[20] G. Turan, Succinct representations of graphs. Discrete Applied Math, 8: 289-294, 1984.

[21] W. Tutte. A census of planar triangulations. Canadian Journal of Mathematics, 14 (1962), pp. 21-38.

[22] W. Tutte. The Enumerative Theory of Planar Graphs. In *A Survey of Computational Theory*, J.N. Srinivasan et al. (Eds.), North-Holland, 1973.