Encoding Quadrilateral Meshes in 2.40 bits per Vertex

Pawel Kosicki
University of Toronto
pav.kosicki@utoronto.ca

Asish Mukhopadhyay
University of Windsor
asishm@uwindsor.ca

Abstract

In this note, we show that the encoding scheme discussed in [1] for quadrilateral meshes can be improved to 2.40 bits per vertex, using arithmetic coding. This is a considerable improvement over the current bound of 2.67 bits per vertex.

1. Introduction

Following the publication of Deering's paper [3], geometric compression has become a very active field of research [2], [1]. The emphasis of the research has primarily been on finding efficient schemes for encoding the geometry of meshes that are made up entirely of triangles, or entirely of quadrilaterals (quad, for short), or a mixture of both. The practical significance of this is that it enables one to store and transmit such meshes (over the Internet) efficiently.

In an earlier paper [5], we showed that the encoding scheme discussed in [1] for quadrilateral meshes (quadmesh, for short) can be improved to less than 3 bits per vertex. We also pointed out that an equivalence between the labelling schemes of King et al [2] and of Kronrod & Gotsman [1], improves this further to 2.67 bits per vertex. The same upper bound has also been reported in [2], making an involved use of the CLRES scheme. We left open the question whether this bound can be improved. The main contribution of this paper is to propose a scheme based on arithmetic coding that improves this bound to 2.4006 bits per vertex. This approach was used by Gumhold [6] to improve the CLERS encoding scheme for triangle meshes [7] from 3.67 bits per vertex to 3.552 bits per vertex.

2. Kronrod-Gotsman scheme

The Kronrod-Gotsman scheme [1] generalizes the CLERS labelling scheme of [2] to non-triangular meshes. Their main observation is that as we traverse a mesh (with or without boundary) in depth-first order, the interaction of each polygon with the rest of the mesh can be enumerated in a finite number of ways. For example, for a quadmesh, each quad interacts with the rest of the mesh in one of 13 ways (Fig.1, arrows indicate the current gate) and hence this interaction can be coded in a unique manner. It is easy to enumerate all these cases if we note that each of the remaining three edges of the current quad either belongs to the mesh boundary or doesn't, and so also for the remaining two vertices.

The compression process traverses the mesh in depth-first order, starting with a quad, at least one of whose edges is part of the mesh boundary. For a closed mesh, we can create a boundary by removing one of the polygons or by introducing opposite orientations on one edge. In the following discussion, the term *gate* will mean an edge of a quad that we are currently visiting, and one that it shares with the current mesh boundary.

For example, if we traverse the quadmesh of Fig.2, starting with the thickly-marked edge on the mesh boundary as the initial gate, and always choose the next gate to be the edge of the current quad that is counterclockwise with respect to the current gate then we get the following output string: $Q_{13}Q_6Q_6Q_5Q_{12}Q_{12}Q_6Q_6Q_{12}Q_{12}Q_9Q_1Q_6Q_1$.

3. Encoding scheme with less than 3.0 bpv

Kronrod & Gotsman [1] proposed a prefix-free variable length encoding scheme for such a string that needs at most 3.5 bits per quad. We show that this can be improved to less than 3 bits per vertex.

As we process a quad, we introduce new edges and vertices. These new edges and new vertices are free edges and vertices that become part of the mesh boundary when we process and remove the current quad. An edge or vertex of a quad is *free* if it doesn't belong to the mesh boundary. Table 1 summarizes this information for each of the thirteen types of quad that is encountered during mesh traversal.

The following observation about the compression process is significant and, along with equations (2-4), underlies the coding scheme of Table 2.

Observation 1 *In a manifold mesh, a quad of type* Q_5, Q_{10}, Q_{12},*or* Q_{13} *is never followed by a quad of type*

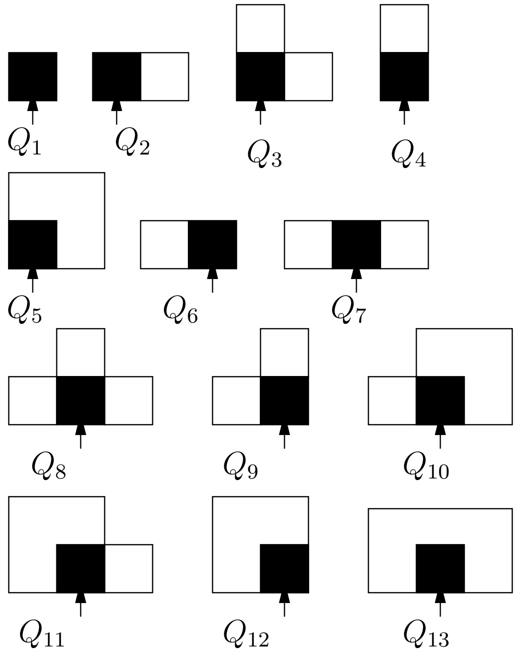

Figure 1. *Interaction of a quad with the mesh*

Q_1, Q_2, Q_3, Q_4, Q_5, *if while traversing the mesh we choose the next gate to be situated counterclockwise with respect to the present one.*

We claim that the coding scheme of Table 2 has an upper bound of 3.0 bits per vertex. For a proof assume that we have a quadmesh homeomorphic to a sphere. Let $|E|$, $|V|$ and $|Q|$ be the count of its edges, vertices and quads respectively. Since each edge is shared by exactly two quads, $|E| = 2|Q|$. Combining this with Euler's formula, we get

$$|V| = |Q| + 2 \tag{1}$$

For a very large mesh, $|Q| >> 2$; therefore, $|V| = |Q|$ approximately. Let $|Q_i|$ denote the number of quads of type Q_i and $|Q_{i-j}| = |Q_i| + \ldots + |Q_j|, j > i$. From $|V| = |Q|$ above and Table 2, it follows that

$$2|Q_{13}| + |Q_5| + |Q_{10-12}| \quad = \quad |Q| \tag{2}$$

as the left-hand side counts the number of vertices in the quadmesh.

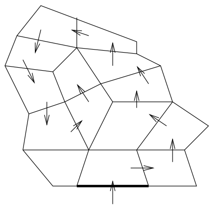

Figure 2. *Traversing a quadmesh*

Quad type	new edges	new vertices
Q_1	0	0
Q_2	1	0
Q_3	2	0
Q_4	1	0
Q_5	2	1
Q_6	1	0
Q_7	2	0
Q_8	3	0
Q_9	2	0
Q_{10}	3	1
Q_{11}	3	1
Q_{12}	2	1
Q_{13}	3	2

Table 1. *Mesh-Quad Interactions*

Again, as $|V| = |Q|$ (approximately), the number of quads which have two free vertices must be equal to the number of quads which have no free vertices. Thus from Table 1, it follows that,

$$|Q_{13}| \quad = \quad |Q_{1-4}| + |Q_{6-9}| \tag{3}$$

Each path from the root in a quad spanning tree ends in a quad of type Q_1 (a leaf node), and each branch begins either at the root gate or at a quad of type Q_3 or of type Q_{7-11}. With each quad of type Q_3, Q_7, Q_9, Q_{10}, and Q_{11}, one additional quad of type Q_1 is associated. With each quad of type Q_8, two additional quads of type Q_1 are associated. Therefore, we have the following constraint, based on a generalization of the formula that in a binary tree, the number of leaf nodes is one more than the nodes of degree 2.

$$|Q_3| + |Q_{7-11}| + |Q_8| = |Q_1| - 1 \tag{4}$$

Using the constraint of equation (3), we can group each of quad of types Q_{1-4} and Q_{6-9} with a quad of type Q_{13} as shown in Table 3.

Thus the grouping of a quad of each of the types Q_{1-9} with a quad of type Q_{13} yield an average bit count of at most 3. We next use the constraint of equation (4) to refine this analysis even further. This constraint implies that $|Q_{7-11}| + |Q_8| < |Q_1| - 1$. Therefore, each of the quad-types from Q_{7-11} and Q_8 can be associated with at most one quad of type Q_1.

Since quads of type Q_3 have been taken care of, we don't have to find groups for these. Since each quad of type Q_1 has been grouped with a quad of type Q_{13} already and one quad of type Q_1 is associated with each one of the quad types Q_7, Q_9, Q_{10}, Q_{11}, while two quads of type

Encoding	Current Quad	Next Quad	Code	Num. of bits
Quad started with Q_{6-13}	Q_6	Q_{1-5}	11111	5
		Q_{6-13}	11110	5
	Q_7	Q_{1-5}	11101	5
		Q_{6-13}	11100	5
	Q_8	Q_{1-5}	11011	5
		Q_{6-13}	11010	5
	Q_9	Q_{1-5}	11001	5
		Q_{6-13}	11000	5
	Q_{10}	Q_{6-13}	10111	5
	Q_{11}	Q_{1-5}	10110	5
		Q_{6-13}	10101	5
	Q_{12}	Q_{6-13}	100	3
	Q_{13}	Q_{6-13}	0	1
Quad started with Q_{1-5}	Q_1	Q_{1-5}	00	2
		Q_{6-13}	01	2
	Q_2	Q_{1-5}	1100	4
		Q_{6-13}	1101	4
	Q_3	Q_{1-5}	1010	4
		Q_{6-13}	1011	4
	Q_4	Q_{1-5}	1000	4
		Q_{6-13}	1001	4
	Q_5	Q_{6-13}	111	3

Table 2. *Coding Scheme*

Q_1's are associated with a quad of type Q_8, we need to associate a single quad-type group (Q_1, Q_{13}) with each one of (Q_7, Q_{13}), (Q_9, Q_{13}), Q_{10}, and Q_{11}. Futher, we need to associate two groups of (Q_1, Q_{13}) with one group of (Q_8, Q_{13}). The grouping details are shown in Table 4.

The above grouping ensures that $Q_7, Q_8, Q_9, Q_{10}, Q_{11}$ can be grouped to achieve an upper bound of at most 3 bits per vertex. Quad-types Q_5 and Q_{12} do not need to be grouped, since these are already assigned 3 bits each. We conclude that quadmesh connectivity can be encoded in less than 3 bits per vertex. Table 5 summarizes the final groupings.

From Table 3, the total cost of the encoding is

$$3|Q| - (Q_2| + |Q_3| + |Q_4|) - 3(|Q_7| + |Q_9|) - 6|Q_8| - |Q_{10}| - |Q_{11}| - 3|Q_3| - 3.$$

Since $|V| = |Q| + 2$ (exactly) for a quadmesh, the total cost is therefore guaranteed to be less than 3 bits per vertex.

4. Improving the upper bound to 2.67 bpv

Table 6 shows the connection between the labelling schemes of Kronrod-Gotsman[KG] and King et al[KRS] This correspondence is obtained by noticing that in the scheme of King et al [2] a quad is implicitly split by a diagonal into two triangles so that the next gate is situated counterclockwise with respect to the current one.

KG	KRS	KG	KRS
Q_1	LE	Q_8	SS
Q_2	LL	Q_9	SR
Q_3	LS	Q_{10}	SC
Q_4	LR	Q_{11}	CS
Q_5	LC	Q_{12}	CR
Q_6	LE	Q_{13}	CC
Q_7	SL		

Table 6. *Labelling scheme correspondence*

From the above table of equivalence of labels, we can obtain an encoding scheme that uses less than 2.67 bits per vertex, using the encoding scheme in [2].

5. New encoding scheme

The new encoding scheme, based on arithmetic encoding, is inspired by the work of [6]. Given the probablity distribution of a message source, in arithmetic coding a source

Quad	bits in Quad	Quad	bits in Quad	Average bits	$\leq$ 3??
Q_1	2	Q_{13}	1	1.5	Yes
Q_2	4	Q_{13}	1	2.5	Yes
Q_3	4	Q_{13}	1	2.5	Yes
Q_4	4	Q_{13}	1	2.5	Yes
Q_6	5	Q_{13}	1	3.0	Yes
Q_7	5	Q_{13}	1	3.0	Yes
Q_8	5	Q_{13}	1	3.0	Yes
Q_9	5	Q_{13}	1	3.0	Yes

Table 3. *Code bits analysis for Q_1 to Q_9*

Group	Total bits	Group	Total bits	Average bits	$\leq$ 3??
(Q_7, Q_{13})	6	(Q_1, Q_{13})	3	2.25	Yes
(Q_8, Q_{13})	6	$(Q_1, Q_{13}, Q_1, Q_{13})$	6	2.0	Yes
(Q_9, Q_{13})	6	(Q_1, Q_{13})	3	2.25	Yes
Q_{10}	5	(Q_1, Q_{13})	3	2.67	Yes
Q_{11}	5	(Q_1, Q_{13})	3	2.67	Yes

Table 4. *Code bits analysis for Q_7 to Q_{11}*

ensemble is represented by a subinterval of the unit interval [0,1). If s is the size of this subinterval, the number of bits required to encode the source ensemble of n messages is $-logs = \Sigma_{\iota=1}^{n} - log_2(v_\iota)$ where v_ι is the probability of occurrence of the ιth message. This is exactly the entropy of the source. For example, if the source ensemble $\{a, b, c, d, \#\}$ has the probabilities $\{0.2, 0.4, 0.1, 0.2, 0.1\}$ then the message ensemble $aadb\#$ is represented by the interval $[0.03248, 0.0328)$, when messages $a, b, c, d, \#$ represent the intervals [0,0.2), [0.2,0.6), [0.6,0.7), [0.7,0.9), [0.9,1) respectively. For more details see [8]. .

6. More constraints on the quad string

First of all, we assume that all the Q-symbols occur with probability v_τ, barring Q_1 and Q_{13}, which occur with probabilities $7v_\tau$ and $14v_\tau$, respectively, as can be seen from equations (3) and (4). Next, we note that each of the 13 Q-symbols modifies the length of the mesh border in one of these ways: increase it by 2, keep it the same, decrease it by 2 or by 4, or split it into two submeshes with borders of unknown length. The effects on the length of the mesh boundary of the individual Q-symbols are shown in Table 7 below.

While the constraint of Observation 1 was the cornerstone of the 3.0 bits per vertex upper bound, to improve this bound we have to think of additional constraints. This comes in the form of the *minimum* currently-allowable mesh-border length. For example, we can't have a quad of type Q_1 if the current mesh-border is longer than 4. At the start, this is always 4 since every non-degenerate mesh

Symbol	Effect	Symbol	Effect
Q_1	-4	Q_8	?
Q_2	-2	Q_9	?
Q_3	?	Q_{10}	?
Q_4	-2	Q_{11}	?
Q_5	0	Q_{12}	0
Q_6	-2	Q_{13}	+2
Q_7	?		

Table 7. *Effect of Q-symbols on border length*

is made up of at least one quad; the probability of this happening is 1. From this step on, the minimum mesh-border length can either increase to 6 (if the next Q-symbol is a Q_{13}) or remain at a minimum of 4 (because it could either be 'reduced' by $Q_{2,4,6}$, but the minimum length would remain as 4; it could be split by $Q_{3,7-11}$ or terminated by Q_1, in which case the new mesh would also have a border length of at least 4, and finally $Q_{5,12}$ would cause the border to remain the same). Let C stand for one of the quad types: Q_5, Q_{10}, Q_{12}, Q_{13}. We therefore have the equation

$$\mathbf{1} \quad = \quad 18v_\tau + 14 * \mathbf{1_{6,C}}v_\tau, \qquad (5)$$

where the *conditional unity* $\mathbf{1_{i,C}}$ indicates that the preceding quad is of type C, while the *conditional unity* $\mathbf{1_C}$ indicates the preceding quad is not of type C. Continuing this argument we obtain the equations of Table 8.

Let us clarify a couple of cases: $\mathbf{1_{10,C}}$ and $\mathbf{1_N}$. $\mathbf{1_{10,C}}$ can turn into $\mathbf{1_8}$ (if a Q_6 occurs, which has a probability of v_τ), into $\mathbf{1_{10,C}}$ (if we have a Q_{12}, which also has a probability of v_τ), into $\mathbf{1_{12,C}}$ (if a Q_{13} occurs, which has a prob-

Grouping	Total Cost	quads in group	Amortized Cost	bits saved	occurrences of this group
(Q_2, Q_{13}) or (Q_3, Q_{13}) or (Q_4, Q_{13})	5	2	2.5	1	$\|Q_2\| + \|Q_3\| + \|Q_4\|$
$(Q_7, Q_{13}, Q_1, Q_{13})$ or $(Q_9, Q_{13}, Q_1, Q_{13})$	9	4	2.25	3	$\|Q_7\| + \|Q_9\|$
$(Q_8, Q_{13}, Q_1, Q_{13}, Q_1, Q_{13})$	12	6	2.0	6	$\|Q_8\|$
(Q_{10}, Q_1, Q_{13})	8	3	2.67	1	$\|Q_{10}\|$
(Q_{11}, Q_1, Q_{13})	8	3	2.67	1	$\|Q_{11}\|$
remaining(Q_1, Q_{13})	3	2	1.5	3	$\|Q_3\|$+1
Q_5	3	1	3	0	$\|Q_5\|$
(Q_6, Q_{13})	6	2	3	0	$\|Q_6\|$
Q_{12}	3	1	3	0	$\|Q_{12}\|$

Table 5. *Amortization Analysis*

cond	follow	equation
4	Q_{1-13}	$\mathbf{1} = (18 + 14 * \mathbf{1_{6,C}})v_\tau$
6,C	Q_{6-13}	$\mathbf{1_{6,C}} = (\mathbf{1} + \mathbf{1_{6,C}} + 14 * \mathbf{1_{8,C}} + 5)v_\tau$
8,C	Q_{6-13}	$\mathbf{1_{8,C}} = (\mathbf{1_6} + \mathbf{1_{8,C}} + 14 * \mathbf{1_{10,C}} + 5)v_\tau$
10,C	Q_{6-13}	$\mathbf{1_{10,C}} = (\mathbf{1_8} + \mathbf{1_{10,C}} + 14 * \mathbf{1_{12,C}} + 5)v_\tau$
.	.	.
.	.	.
.	.	.
i, C	Q_{6-13}	$\mathbf{1_{i,C}} = (\mathbf{1_{i-2}} + \mathbf{1_{i,C}} + 14 * \mathbf{1_{i+2,C}} + 5)v_\tau$
.	.	.
.	.	.
.	.	.
N, C	Q_{6-13}	$\mathbf{1_{N,C}} = (\mathbf{1_{N-2}} + \mathbf{1_{N,C}} + 5)v_\tau$
6	Q_{2-13}	$\mathbf{1_6} = (3 + 2 * \mathbf{1_{6,C}} + 14 * \mathbf{1_{8,C}} + 6)v_\tau$
8	Q_{2-13}	$\mathbf{1_8} = (3 * \mathbf{1_6} + 2 * \mathbf{1_{8,C}} + 14 * \mathbf{1_{10,C}} + 6)v_\tau$
10	Q_{2-13}	$\mathbf{1_{10}} = (3 * \mathbf{1_8} + 2 * \mathbf{1_{10,C}} + 14 * \mathbf{1_{12,C}} + 6)v_\tau$
.	.	.
.	.	.
.	.	.
i	Q_{2-13}	$\mathbf{1_i} = (3 * \mathbf{1_{i-2}} + 2 * \mathbf{1_{i,C}} + 14 * \mathbf{1_{i+2,C}} + 6)v_\tau$
.	.	.
.	.	.
.	.	.
N	Q_{2-13}	$\mathbf{1_N} = (3 * \mathbf{1_{N-2}} + 2 * \mathbf{1_{N,C}} + 6)v_\tau$

Table 8. *Conditional unity table*

ability of $14v_\tau$) and finally into a mesh of unknown length if we have a Q_{7-11} which have a combined probability of $5v_\tau$. If we did not define N to be the maximum allowable border length, we would have an unsolvable infinite set of equations. We therefore say that the border length cannot exceed N. The case of $\mathbf{1_N}$ therefore can change into $\mathbf{1_{N-2}}$ (if $Q_{2,4,6}$ occur) with a probability of $3v_\tau$, into $\mathbf{1_{N,C}}$ (in the case of $Q_{5,12}$), into $\mathbf{1_{N+2,C}}$ with a probability of 0 (since N is the maximum border length) or into submeshes of unknown lengths if we encounter the Q-symbols $Q_{3,7-11}$.

We proceded to solve these equations for various values of N using Computer Algebra packages Maple and Matlab. The results are shown in Table 9 below, from which it is evident that v_τ converges to a value less than 0.039.

N	v_τ	bits required
12	0.03984	2.3698
16	0.03926	2.3910
20	0.03909	2.3972
24	0.03903	2.3995
30	0.03901	2.4002
54	0.03900093	
100	0.03900092	
	0.03900	2.4006

Table 9. *Bits for different N*

We summarize the dicussion above in the following theorem.

Theorem: A quadmesh can be encoded in no more than 2.4006 bits per vertex.

7. Conclusions

In this note, we have shown that a scheme proposed by Kronrod & Gotsman [1] for quadmesh encoding can be improved to have an upper bound of less than 2.4006 bits per vertex. We supsect that this bound can be lowered still further.

References

[1] Boris Kronrod and Craig Gotsman. *Efficient coding of non-triangular mesh connectivity*. In Proc. 16th European Workshop on Computational Geometry, pp. 24-26, 2000.

[2] Davis King, Jarek Rossignac and Andrzej Szymczak. *Connectivity compression for irregular quadrilateral meshes*. GVU Tech Report GVU-GIT-99-36.

[3] M. Deering. *Geometry Compression*. In Proc. ACM SIGGRAPH'95, pp. 13-20, August 1995.

[4] T. H. Cormen, C. E. Leiserson and R. L. Rivest. *Introduction to Algorithms*, Prentice-Hall, 1998.

[5] Asish Mukhopadhyay and Quanbin Jing. *Encoding Quadrilateral Meshes*. In Proc. 15th CCCG, Dalhousie University, Halifax, pp. 60-63, 2003.

[6] Stefan Gumhold. *New bounds on the encoding of planar triangulations*, WSI-2000-1, ISSN 0946-3852, March 9, 2000.

[7] Davis King and Jarek Rossignac. *Guaranteed 3.67V bit encoding of planar triangle meshes*. In Proc. of the 11th CCCG, pp. 146-149,1999.

[8] Debra A. Lelewer and Daniel S. Hirchberg. *Data Compression*. ACM Computing Surveys, Vol. 19, No. 3, pp. 261-296, 1987.