Efficient enumeration of 3-manifold triangulations

Benjamin A. Burton

Abstract

Triangulations describe how different low-dimensional topological spaces can be constructed from simple building blocks. Forming an exhaustive list of all small triangulations is difficult, typically involving a slow computer-based search. We present techniques for improving the efficiency of such searches, thereby allowing more extensive lists to be constructed.

1 Introduction

Low-dimensional topologists take a particular interest in the structures of spaces of small dimension. An *n-manifold* is essentially a topological space in which each point has an n-dimensional neighbourhood. So, for instance, an infinite plane or the surface of a sphere are both 2-manifolds, whereas the universe in which we live is a 3-manifold.

Triangulations offer a convenient method for representing n-manifolds using n-dimensional simplices as basic building blocks. For instance, 2-manifolds can be constructed by gluing together triangles along their edges, and 3-manifolds can likewise be formed by gluing together tetrahedra along their faces.

Much of the current activity in low-dimensional topology focuses upon the classification of 3-manifolds. To this end, an exhaustive list of all 3-manifold triangulations of a particular size and type can be a useful resource. Such a list is called a *census*, and can provide a rich body of examples for testing and analysis as well as offering insight into the structures of the most efficient triangulations.

A number of such censuses are described in the literature [1, 2, 5, 6, 7, 8, 10, 12]. Typically a census is formed through an exhaustive computer search over the possible ways in which tetrahedron faces can be joined together.[1] Such searches are exceptionally slow however, and therefore only limited results have been found to date.

In order to improve the efficiency of such computer searches, most authors use mathematical techniques to restrict the search space and thus speed up the census generation. Here we present a series of such techniques that were introduced for the seven-tetrahedron closed non-orientable census [5]. We focus in particular on the use of graphs describing which tetrahedron faces are joined to which others.

Section 2 describes the overall structure of a typical census algorithm. In Section 3 we introduce the graphs discussed above and present the main results. Finally Section 4 incorporates these results into the census algorithm and examines how effective these improvements are in practice. For a full exposition of this material, the reader is referred to [4].

[1] The exceptions are the non-orientable censuses of Amendola and Martelli [1, 2], who use theoretical techniques but do not enumerate the different triangulations of each 3-manifold.

2 The Census Algorithm

Before presenting the census algorithm, we make some additional comments on the formation of triangulations from individual tetrahedra. When joining two tetrahedron faces together as part of a 3-manifold triangulation, there are six possible ways in which this join can take place. These six ways correspond to the six rotations and reflections of the equilateral triangle.

It is important to note that this joining of faces does not need to take place in 3-dimensional space, i.e., triangulations are not subject to "physical constraints". The tetrahedra can simply be pictured as individual objects whose faces are linked in pairs – if you travel through the 3-manifold and exit one tetrahedron along some face, you immediately enter some other tetrahedron along another. This flexibility allows us to use relatively simple triangulations to represent a rich variety of different 3-manifolds.

Example 2.1. Consider the two-tetrahedron triangulation illustrated in Figure 1 (this triangulation represents the 3-manifold $S^2 \times S^1$). The two faces BAC and ACD are joined to each other so that vertices B, A and C correspond to vertices A, C and D respectively. The two faces QPR and PRS are joined to each other so that vertices Q, P and R correspond to vertices P, R and S respectively. In a similar fashion, faces ABD and PQS are joined together and faces BCD and QRS are joined together.

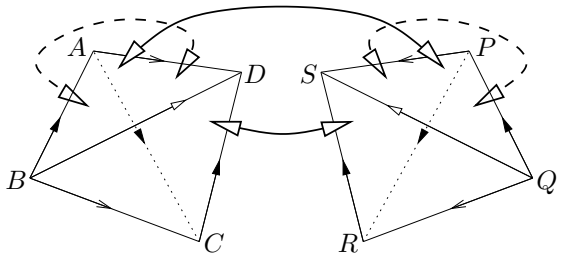

Figure 1: A two-tetrahedron triangulation of $S^2 \times S^1$

As a result of these face gluings, the six edges BA, AC, CD, QP, PR and RS come together to form a single edge of the triangulation. Likewise the four edges AD, BC, PS and QR become one, and the two edges BD and QS become one. These edge identifications are graphically depicted in Figure 1 using identical arrowheads for identified edges.

We pause now to formally lay down our census constraints. As in most previous censuses of 3-manifold triangulations, we restrict our attention to only the simplest triangulations of the simplest 3-manifolds as follows.

Definition 2.2 (Census Triangulation)**.** We define a *census triangulation* to be a 3-manifold triangulation satisfying the following additional constraints.

- The triangulation must be closed, i.e., the 3-manifold it represents must be compact and have no boundary. In a 2-dimensional analogy, this constraint would rule out a plane (which is infinite) or a disc (which has a boundary circle) but would allow a sphere or a torus.
- The triangulation must represent a $\mathbb{P}^2$-irreducible 3-manifold. $\mathbb{P}^2$-irreducibility is like primality; it allows us to restrict our attention to the simplest 3-manifolds from which larger 3-manifolds can be constructed.
- The triangulation must be minimal, i.e., there must be no triangulation of an equivalent 3-manifold formed from strictly fewer tetrahedra.

The algorithm for enumerating census triangulations can be split into three main components as described below. This is a fairly natural breakdown, and all enumeration algorithms described in the literature follow this general pattern.

Algorithm 2.3 (Census Algorithm)**.** To enumerate all census triangulations formed from n tetrahedra, we perform the following steps.

1. Given n tetrahedra, enumerate all possible ways in which the $4n$ tetrahedron faces can be partitioned into $2n$ pairs. We refer to such a partition as a *face pairing*.
2. For each face pairing, enumerate all possible rotations and reflections with which each pair of faces can be joined.

   In general each pair of faces can be joined according to one of six possible rotations or reflections, or one of three if only orientable triangulations are to be considered. Thus for each face pairing we must search through approximately 6^{2n} possibilities, or 3^{2n} for just orientable triangulations.

   In practice this search is pruned somewhat. For instance, a common technique is to prune the search tree where it becomes apparent that an edge of low degree will be present in the final triangulation [4, 6, 12]. Other pruning techniques are also available, frequently varying with different census constraints.
3. For each resulting triangulation that is constructed, test whether it satisfies the full set of census constraints (such as minimality and $\mathbb{P}^2$-irreducibility). If it does, include it in the final list of results.

The efficiency of step 3 is difficult to measure, since it often involves a combination of computation and human analysis. We therefore focus our efficiency analysis on the generation of triangulations as described in steps 1 and 2.

Step 1 is exceptionally fast – for the six tetrahedron non-orientable census that consumed fifteen hours of processor time (see Table 2), the generation of face pairings took under a hundredth of a second. We see then that the vast bulk of processor time is spent in step 2 selecting rotations and reflections for each face pairing. It is in step 2 then that we seek to make the strongest improvements.

3 Face Pairing Graphs

Our strategy for improvement the efficiency of Algorithm 2.3 is twofold.

- We derive constraints that a face pairing of a census triangulation must satisfy, as seen in Theorem 3.4. We then throw away any face pairings generated in step 1 of the census algorithm that do not satisfy these constraints.

  Since the time taken to generate face pairings is negligible, it follows that if we can eliminate $p\%$ of our face pairings in this way, we should expect to eliminate approximately $p\%$ of the running time of the overall algorithm.
- We use properties of face pairings to derive properties of the rotations and reflections used in the actual face gluings, as seen in Theorem 3.6. For each face pairing that remains to be processed, we can use these properties to restrict the search described in step 2 of the census algorithm, in some cases making marked improvements to the resulting time complexity.

The results that we derive are best expressed in terms of face pairing graphs. A face pairing graph is simply a convenient visual representation of a face pairing as follows.

Definition 3.1 (Face Pairing Graph)**.** Let T be a 3-manifold triangulation formed from n tetrahedra. The *face pairing graph* of T is the multigraph on n vertices representing the n tetrahedra of T, with $2n$ edges representing the $2n$ pairs of tetrahedron faces that are joined together.

Example 3.2. Recall the two-tetrahedron triangulation of $S^2 \times S^1$ presented in Example 2.1. The corresponding face pairing graph is illustrated in Figure 2. The two vertices correspond to the two tetrahedra. The two loops represent the fact that each tetrahedron has two of its faces joined together, and the double edge between the two vertices represents the fact that two faces of the first tetrahedron are joined with two faces of the second.

Figure 2: The face pairing graph corresponding to Figure 1

It is straightforward to observe that the face pairing graph of any census triangulation is a connected multigraph whose vertices each have degree four. Figure 3 illustrates all such multigraphs on four vertices.

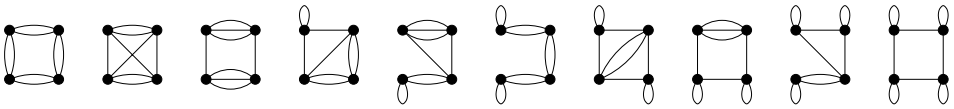

Figure 3: All possible face pairing graphs on four vertices

Definition 3.3 (Chain)**.** A *chain of length* k is a series of k double edges joining $k+1$ distinct vertices in a linear fashion, as illustrated in the left hand diagram of Figure 4. A *one-ended chain* is a chain with a loop appended to one end, as illustrated in the right hand diagram of Figure 4. A solitary loop is considered to be a one-ended chain of length zero.

Figure 4: A chain of length four and a one-ended chain of length three

Our first improvements to the census algorithm involve identifying particular families of subgraphs within face pairing graphs, as seen in the following result.

Theorem 3.4. *Let G be the face pairing graph of a census triangulation containing ≥ 3 tetrahedra. Then G must satisfy the following constraints.*

- *G does not contain a triple edge, i.e., three edges joining the same two distinct vertices as illustrated in the left hand diagram of Figure 5.*
- *G does not contain a pair of one-ended chains joined by a single edge as illustrated in the central diagram of Figure 5, unless this single edge is in fact part of a double edge in G.*
- *G does not contain a one-ended chain with a double handle as illustrated in the right hand diagram of Figure 5. A double handle is simply a triangle with one double edge, attached to the opposite end of the chain from the loop.*

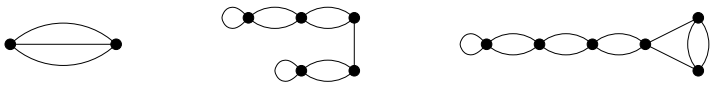

Figure 5: Subgraphs that should not occur within a face pairing graph

Proof. This result is proven in two stages, details of which can be found in [4]. We begin by establishing a number of forbidden properties that a census triangulation must not have. Examples include edges of low degree, or faces joined together to form small structures such as spheres or cones.

We then perform a large case analysis covering all possible ways in which tetrahedra may be joined together according to each type of face pairing graph. For instance, in proving that G cannot contain a triple edge, we examine all 6^3 ways in which two tetrahedra may be joined together along three different faces (corresponding to the two graph vertices joined by three different edges). In each case we derive one of the forbidden properties noted above. □

The previous result describes structures that cannot occur at all within a face pairing graph. We turn now to face pairing graphs that do actually correspond to triangulations in our census. In particular we focus upon one-ended chains within face pairing graphs, and examine the structure that a one-ended chain imposes upon such a triangulation.

We find that one-ended chains have strong links to layered solid tori, which are structures found within a remarkable number of census triangulations [5, 12]. Layered solid tori and related constructs have been described by Jaco and Rubinstein [9], Martelli and Petronio [11] and Matveev [12].

Definition 3.5 (Layered Solid Torus)**.** A *layered solid torus* is a triangulation formed according to the following construction. We begin with the one-tetrahedron triangulation illustrated in Figure 6, where faces ABC and BCD are joined together and faces ABD and ACD are left alone for now.

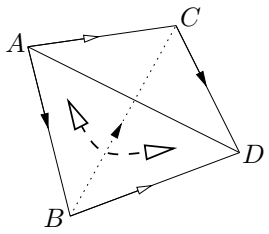

Figure 6: A one-tetrahedron layered solid torus

We may then optionally perform any number of layerings as follows. At each stage there are always two faces not yet joined to anything (such as faces ABD and ACD above). To perform a layering, we select some edge e lying between these two faces, take a new tetrahedron Δ and then join two adjacent faces of Δ to these two faces without any rotations or reflections, as illustrated in Figure 7. The two remaining faces of Δ become our two new unmatched faces, ready for the next layering if we should choose to perform one.

It can be shown that for each layering there are three possible edges e to choose from, and that this construction always yields a triangulation of the solid torus.

We see then that the face pairing graph of a layered solid torus is in fact a one-ended chain. This relationship goes both ways as seen in the following result.

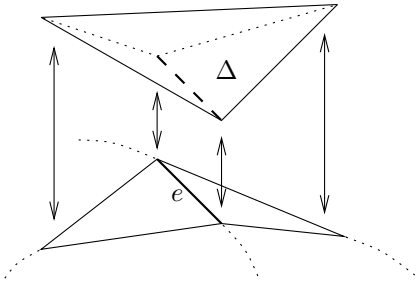

Figure 7: Performing a layering

Theorem 3.6. *Let G be the face pairing graph of a census triangulation T containing ≥ 3 tetrahedra. If G contains a one-ended chain, then the tetrahedra of T corresponding to this one-ended chain form a layered solid torus in T.*

Proof of Theorem 3.6. This proof follows a similar structure to the proof of Theorem 3.4. Specifically we establish a series of forbidden properties of census triangulations. We then perform a case analysis over the possible ways in which a one-ended chain can correspond to actual tetrahedra joined together along their faces.

The case analysis here is refreshingly simpler than in Theorem 3.4. Once more the reader is referred to [4] for details. □

4 Improving the Algorithm

We consider now the application of Theorems 3.4 and 3.6 to the enumeration of 3-manifold triangulations as described in Algorithm 2.3.

As outlined at the beginning of Section 3, we can eliminate any face pairings whose graphs do not satisfy the properties of Theorem 3.4 after running step 1 of the census algorithm. Table 1 illustrates for small numbers of tetrahedra how many face pairings can be eliminated in this way. All counts are given up to graph isomorphism and were obtained using *Regina* [3], a freely available software package for 3-manifold topologists.

The final column of the table lists the running time taken by *Regina* to enumerate the face pairings, as measured on a single 1.2GHz Pentium III processor. Note that 0:00 simply indicates a running time of less than half a second.

Tetrahedra	**Total**	**Eliminate**	**Keep**	**Time (*h:mm:ss*)**
3	4	2 (50%)	2 (50%)	0:00
4	10	6 (60%)	4 (40%)	0:00
5	28	16 (57%)	12 (43%)	0:00
6	97	58 (60%)	39 (40%)	0:00
7	359	221 (62%)	138 (38%)	0:01
8	1 635	997 (61%)	638 (39%)	0:05
9	8 296	4 930 (59%)	3 366 (41%)	0:44
10	48 432	27 681 (57%)	20 751 (43%)	7:21
11	316 520	172 691 (55%)	143 829 (45%)	1:20:48

Table 1: Frequency of face pairing graphs that can be eliminated

We see from Table 1 that in each case a little over half of the possible face pairings can be eliminated. Although this does not reduce the time complexity of the algorithm, we do expect the running time to be reduced by more than 50%, a welcome improvement for a census that may take months or years to complete.

It is worth noting that for ≤ 4 tetrahedra, Theorem 3.4 perfectly divides the face pairings into those that lead to a census triangulation and those that do not. For ≥ 5 tetrahedra however this is no longer true – there are graphs that are not eliminated but which do not yield any triangulations in the final census results. Thus further research can be undertaken to allow the elimination of even more face pairings.

Consider now step 2 of Algorithm 2.3, in which we search through all possible rotations and reflections for each face pairing. Recall that it is here that almost all of the algorithm's running time is concentrated. We can use Theorem 3.6 to redesign this search as follows.

Before the search begins, we identify all one-ended chains within the face pairing graph. For each one-ended chain we search through all possible layered solid tori of the appropriate size. Finally, for each edge of the graph not belonging to a one-ended chain, we search through all six possible rotations and reflections for the corresponding face gluing as before.

The resulting improvement in running time should be substantial, as suggested by the following rough calculations. Consider a one-ended chain on k vertices containing $2k-1$ edges. In the original algorithm we must search through all rotations and reflections for $2k-1$ different face gluings, covering up to $6^{2k-1} = \frac{1}{6}36^k$ possibilities ($3^{2k-1} = \frac{1}{3}9^k$ if we seek only orientable triangulations), although this number will be reduced due to pruning.

The redesigned algorithm however reduces this search to all layered solid tori on k vertices. It can be shown that this offers just 2^k possibilities, a significant improvement on the original estimates of $\frac{1}{6}36^k$ and $\frac{1}{3}9^k$.

Table 2 measures the resulting improvements in running times for several experimental censuses of triangulations. The censuses are divided according to whether only orientable triangulations or only non-orientable triangulations are sought. All running times are displayed as *h*:*mm*:*ss*, are measured on a single 1.2GHz Pentium III processor and were obtained using the program *Regina* [3]. As expected the most significant improvements are obtained in the non-orientable censuses, in the six-tetrahedron case reducing the running time from five weeks to just over fifteen hours.

	Orientable		**Non-Orientable**	
Tetrahedra	**Old Time**	**New Time**	**Old Time**	**New Time**
3	0:00	0:00	0:02	0:00
4	0:03	0:01	3:03	0:06
5	2:28	0:51	6:01:53	9:19
6	2:49:29	56:00	5 weeks	15:08:43

Table 2: Algorithm running times before and after improvements

References

[1] Gennaro Amendola and Bruno Martelli, *Non-orientable 3-manifolds of small complexity*, Topology Appl. **133** (2003), no. 2, 157–178.

[2] _____, *Non-orientable 3-manifolds of complexity up to 7*, Topology Appl. **150** (2005), no. 1-3, 179–195.

[3] Benjamin A. Burton, *Regina: Normal surface and 3-manifold topology software*, `http://regina.sourceforge.net/`, 1999–2009.

[4] ———, *Face pairing graphs and 3-manifold enumeration*, J. Knot Theory Ramifications **13** (2004), no. 8, 1057–1101.

[5] ———, *Structures of small closed non-orientable 3-manifold triangulations*, J. Knot Theory Ramifications **16** (2007), no. 5, 545–574.

[6] Patrick J. Callahan, Martin V. Hildebrand, and Jeffrey R. Weeks, *A census of cusped hyperbolic 3-manifolds*, Math. Comp. **68** (1999), no. 225, 321–332.

[7] Martin V. Hildebrand and Jeffrey R. Weeks, *A computer generated census of cusped hyperbolic 3-manifolds*, Computers and Mathematics (Cambridge, MA, 1989), Springer, New York, 1989, pp. 53–59.

[8] Craig D. Hodgson and Jeffrey R. Weeks, *Symmetries, isometries and length spectra of closed hyperbolic three-manifolds*, Experiment. Math. **3** (1994), no. 4, 261–274.

[9] William Jaco and J. Hyam Rubinstein, *Layered-triangulations of 3-manifolds*, Preprint, `arXiv:math/0603601`, March 2006.

[10] Bruno Martelli and Carlo Petronio, *Three-manifolds having complexity at most 9*, Experiment. Math. **10** (2001), no. 2, 207–236.

[11] ———, *Complexity of geometric three-manifolds*, Geom. Dedicata **108** (2004), no. 1, 15–69.

[12] Sergei V. Matveev, *Tables of 3-manifolds up to complexity 6*, Max-Planck-Institut für Mathematik Preprint Series (1998), no. 67, available from `http://www.mpim-bonn.mpg.de/html/preprints/preprints.html`.

Department of Mathematics and Statistics
The University of Melbourne
Parkville VIC 3010, Australia
`bab@debian.org`