A Fast Algorithm for Computing Irreducible Triangulations of Closed Surfaces in $\mathbb{E}^d$ and Its Application to the TriQuad Problem

Thiago Lemos[1], Suneeta Ramaswami[2], and Marcelo Siqueira[3]

[1]Programa de Pós-Graduação em Informática, Universidade Federal do Paraná, Curitiba, Brazil
[2]Department of Computer Science, Rutgers University, Camden, NJ, USA
[3]Departamento de Matemática, Universidade Federal do Rio Grande do Norte, Natal, Brazil

October 30, 2014

(Extended Abstract)

Let $\mathcal{S}$ be a compact surface with empty boundary. A classical result from the 1920s by Tibor Radó asserts that every compact surface with empty boundary (usually called a *closed surface*) admits a triangulation [1]. Let e be any edge of a triangulation $\mathcal{T}$ of $\mathcal{S}$. The *contraction* of e in $\mathcal{T}$ consists of contracting e to a single vertex and collapsing each of the two triangles meeting e into a single edge. If the result of contracting e in $\mathcal{T}$ is still a triangulation of $\mathcal{S}$, then e is said to be *contractible*; else it is *non-contractible*. A triangulation $\mathcal{T}$ of $\mathcal{S}$ is said to be *irreducible* if and only if every edge of $\mathcal{T}$ is non-contractible. Barnette and Edelson [2] showed that all closed surfaces have finitely many irreducible surfaces. More recently, Boulch, de Verdière, and Nakamoto [3] showed the same result for compact surfaces with a nonempty boundary. Irreducible triangulations have proved to be an important tool for tackling problems in combinatorial topology, and discrete and computational geometry. The reasons are two-fold. First, all irreducible triangulations of any given compact surface form a "basis" for all triangulations of the same surface. Indeed, every triangulation of the surface can be obtained from at least one of its irreducible triangulations by a sequence of *vertex splittings* [4, 5], where the vertex splitting operation is the inverse of the edge contraction operation. Second, some problems on triangulations can be solved by considering irreducible triangulations only. In particular, irreducible triangulations have been used for proving the existence of geometric realizations (in some $\mathbb{E}^d$) of triangulations of certain surfaces, where $\mathbb{E}^d$ is the d-dimensional Euclidean space [6, 7], for studying properties of diagonal flips on surface triangulations [8, 9, 10, 11], for characterizing the structure of flexible triangulations of the projective plane [12], and for finding lower and upper bounds for the maximum number of cliques in an n-vertex graph embeddable in a given surface [13]. An irreducible triangulation is also "small", as its number of vertices is at most linear in the genus of the surface [14, 3]. However, the number of vertices of all irreducible triangulations of the same surface may vary, while any irreducible triangulation of smallest size (known as *minimal*) has $\Theta(\sqrt{g})$ vertices if the genus g of the surface is positive [15]. The sphere has a unique irreducible triangulation, which is the boundary of a tetrahedron [16]. The torus has exactly 21 irreducible triangulations, whose numbers of vertices vary from 7 to 10 [17]. The projective plane has only two irreducible triangulations, one with 6 vertices and the other with 7 vertices [18]. The Klein bottle has exactly 29 irreducible triangulations with numbers of vertices ranging from 8 to 11 [19]. Sulanke devised and implemented an algorithm for generating all irreducible triangulations of compact surfaces with empty boundary [5]. Using this algorithm, Sulanke rediscovered the aforementioned irreducible triangulations and generated the complete sets of irreducible triangulations of the double torus, the triple cross surface, and the quadruple cross surface. The idea behind Sulanke's algorithm is to generate irreducible triangulations of a surface by modifying the irreducible triangulations of other surfaces of smaller Euler genuses. The modifications include vertex splittings and the addition of handles, crosscaps, and crosshandles. Unfortunately, the lack of a known upper bound on the number of vertex splittings required in the intermediate stages of the algorithm prevented Sulanke from establishing a termination criterion for all surfaces. Furthermore, his algorithm is impractical for surfaces with Euler genus ≥ 5, as his implementation could take centuries to generate the quintuple cross surface on a cluster of computers with an average CPU speed of 2GHz [5]. To the best of our knowledge, no similar algorithm for compact surfaces with a nonempty boundary is known.

In this work, we give an algorithm for a problem closely related to that of Sulanke's: *given any triangulation $\mathcal{T}$ of a compact surface $\mathcal{S}$ with empty boundary, find one irreducible triangulation, $\mathcal{T}'$, of $\mathcal{S}$ from $\mathcal{T}$*. In particular, if the genus g of $\mathcal{S}$ is positive, then we show that $\mathcal{T}'$ can be computed in $\mathcal{O}(gn + g^2)$ time, where n is the number of triangles of $\mathcal{T}$. Otherwise, $\mathcal{T}'$ can be computed in $\mathcal{O}(n)$ time, which is optimal. In either case, the space requirement is in $\Theta(n)$. To the best of our knowledge, the previously best known (time) upper bound is $\mathcal{O}(n \lg n + g \lg n + g^4)$ for the algorithm given by Schipper in [4]. In his complexity analysis, Schipper assumed that g is a constant depending only on $\mathcal{S}$, and thus stated the upper bound as $\mathcal{O}(n \lg n)$. While it is true that g is an intrinsic feature of $\mathcal{S}$, we may have $m \in \Theta(\sqrt{g})$ [15], where m is the number of vertices of $\mathcal{T}$, which implies that $n \in \Theta(g)$ (from Euler's formula). Thus, we state the time bounds in terms of both g and n. Since our algorithm can more efficiently generate *one* irreducible triangulation from any given triangulation of $\mathcal{S}$, we believe that it can be used as a "black-box" by a fast and alternative method (to that of Sulanke's) for generating *all* irreducible triangulations of any given surface. We implemented our algorithm, the algorithm given by Schipper in [4], and a randomized, brute-force algorithm, and then experimentally compared these implementations on triangulations typically found in graphics applications, as well as triangulations specially designed to study the runtime of the three algorithms in worst-case scenarios [20]. Our algorithm outperformed the other two in all case studies, indicating that the key ideas we use to reduce the worst-case time complexity of our algorithm are also effective in the average case and for triangulations typically encountered in practice. Finally, our algorithm for computing irreducible triangulations was recently incorporated into an innovative and efficient solution [21] to the problem of converting a triangulation $\mathcal{T}$ of a closed surface into a quadrangulation with the same set of vertices as $\mathcal{T}$ (known as the *TriQuad problem* [22]). This new solution also takes $\mathcal{O}(gn + g^2)$ time, where n is the number of triangles of $\mathcal{T}$, to produce the quadrangulation if the genus g of the surface is positive. Otherwise, the solution takes $\mathcal{O}(n)$ amortized time. Our solution improves upon the approach of computing a perfect matching on the dual graph of $\mathcal{T}$, for which the best known upper bound is $\mathcal{O}(n \lg^2 n)$ amortized time [23]. In [21], our solution is experimentally compared with two simple greedy algorithms [24, 25] and the approach based on the algorithm in [23]. It outperforms the approaches in [24, 25, 23] whenever n is sufficiently large and $g \ll n$, which is typically the case for triangulations used in computer graphics and engineering applications. We hope that the solution we devised for the TriQuad problem helps increase the practical interest of the computational geometry and topology community in algorithms to compute irreducible triangulations of surfaces.

References

[1] Jean Gallier and Dianna Xu. *A Guide to the Classification Theorem for Compact Surfaces*. Springer-Verlag, 2013.

[2] David W. Barnette and Allan L. Edelson. All 2-manifolds have finitely many minimal triangulations. *Israel Journal of Mathematics*, 67(1):123–128, 1989.

[3] Alexandre Boulch, Éric Colin de Verdière, and Atsuhiro Nakamoto. Irreducible triangulations of surfaces with boundary. *Graphs and Combinatorics*, 29(6):1675–1688, 2013.

[4] Haijo Schipper. Generating triangulations of 2-manifolds. In Hanspeter Bieri and Hartmut Noltemeier, editors, *Computational Geometry: Methods, Algorithms and Applications*, volume 553 of *Lecture Notes in Computer Science*, pages 237–248. Springer, 1991.

[5] Thom Sulanke. Generating irreducible triangulations of surfaces. *CoRR*, arXiv:math/0606687, 2006.

[6] Dan Archdeacon, C. Paul Bonnington, and Joanna A. Ellis-Monaghan. How to exhibit toroidal maps in space. *Discrete & Computational Geometry*, 38(3):573–594, 2007.

[7] C. Paul Bonnington and Atsuhiro Nakamoto. Geometric realization of a triangulation on the projective plane with one face removed. *Discrete & Computational Geometry*, 40(1):141–157, 2008.

[8] Seiya Negami. Diagonal flips of triangulations on surfaces, a survey. *The Yokohama Mathematical Journal*, 47:1–40, 1999.

[9] Carmen Cortés, Clara Grima, Alberto Marquez, and Atsuhiro Nakamoto. Diagonal flips in outer-triangulations on closed surfaces. *Discrete Mathematics*, 254(1-3):63–74, 2002.

[10] Ken-ichi Kawarabayashi, Atsuhiro Nakamoto, and Yusuke Suzuki. N-flips in even triangulations on surfaces. *Journal of Combinatorial Theory, Series B*, 99(1):229–246, 2009.

[11] Yusuke Higuchi, Atsuhiro Nakamoto, Katsuhiro Ota, and Tadashi Sakuma. N-flips in even triangulations on the torus and dehn twists preserving monodromies. *Discrete Mathematics*, 311(13):1128–1135, 2011. Selected Papers from the 22nd British Combinatorial Conference.

[12] Beifang Chen and Serge Lawrencenko. Structural characterization of projective flexibility. *Discrete Mathematics*, 188(1-3):233–238, 1998.

[13] Vida Dujmović, Gašper Fijavž, Gwenaël Joret, Thom Sulanke, and David R. Wood. On the maximum number of cliques in a graph embedded in a surface. *European Journal of Combinatorics*, 32(8):1244–1252, 2011.

[14] Atsuhiro Nakamoto and Katsuhiro Ota. Note on irreducible triangulations of surfaces. *Journal of Graph Theory*, 20(2):227–233, 1995.

[15] Mark Jungerman and Gerhard Ringel. Minimal triangulations on orientable surfaces. *Acta Mathematica*, 145(1):121–154, 1980.

[16] Eduard Steinitz and Hans Radamacher. *Vorlesungen über die Theorie der Polyder*. Springer, 1934.

[17] Serge Lavrenchenko. Irreducible triangulations of the torus. *Journal of Soviet Mathematics*, 51(5):2537–2543, 1990.

[18] David Barnette. Generating the triangulations of the projective plane. *Journal of Combinatorial Theory, Series B*, 33(3):222–230, 1982.

[19] Thom Sulanke. Note on the irreducible triangulations of the klein bottle. *Journal of Combinatorial Theory, Series B*, 96(6):964–972, 2006.

[20] Suneeta Ramaswami and Marcelo Siqueira. A fast algorithm for computing irreducible triangulations of closed surfaces in $\mathbb{E}^d$. *CoRR*, arXiv:1409.6015, 2014.

[21] Thiago Lemos, Suneeta Ramaswami, and Marcelo Siqueira. An experimental comparison of algorithms for converting triangulations of closed surfaces into quadrangulations. (*Manuscript, to be submitted*).

[22] David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Cláudio Silva, Marco Tarini, and Denis Zorin. Quad-mesh generation and processing: A survey. *Computer Graphics Forum*, 32(6):51–76, 2013.

[23] Krzysztof Diks and Piotr Stanczyk. Perfect matching for biconnected cubic graphs in $\mathcal{O}(n \lg^2 n)$ time. In *Proceedings of the 36th Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'10)*, pages 321–333, Germany, January 23-29 2010. Springer-Verlag.

[24] Kari Pulli and Mark E. Segal. Fast rendering of subdivision surfaces. In *Proceedings of the Eurographics Workshop on Rendering Techniques*, pages 61–70, August 27-31 1996.

[25] Luiz Velho. Quadrilateral meshing using 4-8 clustering. In *Proceedings of the Symposium on Mesh Generation and Self-Adaptivity (CILANCE 2000)*, pages 61–64, 2000.