INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

3-Colored Triangulation of 2D Maps

Lucas Moutinho Bueno *Jorge Stolfi*

Technical Report - IC-14-10 - Relatório Técnico

July - 2014 - Julho

3-Colored Triangulation of 2D Maps

Lucas Moutinho Bueno Jorge Stolfi [*]

Abstract

We describe an algorithm to triangulate a general map on an arbitrary surface in such way that the resulting triangulation is vertex-colorable with three colors. (Three-colorable triangulations can be efficiently represented and manipulated by the GEM data structure of Montagner and Stolfi.) The standard solution to this problem is the barycentric subdivision, which produces $4e - 2b$ triangles when applied to a map with e edges, such that b of them are border edges (adjacent to only one face). Our algorithm yields a subdivision with at most $2e - b + 2(2 - \chi)$ triangles, where χ is the Euler Characteristic of the surface; or at most $2e - n - 2b + 4(2 - \chi)$ triangles if all n faces of the map have the same degree. Experimental results show that the resulting triangulations have, on the average, significantly fewer triangles than these upper bounds.

1 Introduction

We describe an algorithm to triangulate a general map M on an arbitrary surface to produce a triangulation T that is 3-vertex-colorable, namely whose vertices can be labeled with the three "colors" $\{0, 1, 2\}$ in such a way that the endpoints of every edge have distinct colors. Vertex-colorable triangulations can be efficiently represented and manipulated by the GEM (Graph-Encoded Manifolds) data structure of Montagner and Stolfi [7, 6], based on the graph class of the same name created by Sóstenes Lins and Arnaldo Mandel [5].

The standard solution to this problem is the *barycentric subdivision* [2], where each edge of M is divided into two new edges and a new vertex, and each face of M is divided into $2d$ triangles, $2d$ new edges, and a new vertex, where d is the degree of the divided face. The procedure also yields a 3-vertex-coloration for M, where each vertex has color 0, 1 or 2 depending on whether it is contained in a vertex, edge, or face of M, respectively. The barycentric subdivision is easily implemented, but always produces a triangulation T with $4e - 2b$ triangles when applied to a map M with e edges.

Here we extend our previous algorithm for transforming an arbitrary triangulation into a colored one [3] to work with general maps with faces of any degree. This new algorithm is equivalent to the former one if the input is a triangulation. Our algorithm is considerably more efficient than barycentric subdivision: it yields a triangulation with at most $2e - b + 2(2 - \chi)$ triangles, where χ is the Euler characteristic of the surface. If the input map M

[*]Institute of Computing, University of Campinas, 13081-970 Campinas, SP, Brazil. Financial support from FAPESP, process 2012/14698-7

has n faces and all of them have the same degree D, the resulting triangulation has at most $2e - n - 2b + 4(2 - \chi)$ $(= (D-1)n - b + 4(2-\chi))$ triangles. These are only upper bounds. Experiments with certain classes of random maps (described in Section 7) shows that average size of the output is lower than the upper bound, especially for maps with faces with larger average degree.

A naive algorithm for the problem considered here would be (1) randomly triangulate the faces of M and (2) apply our previous algorithm to the resulting triangulation. The triangulation resulting from step (1) would have $L_0(M) = 2e - 2n - b + k$ triangles, where k is the number of faces of M with degree 2. Therefore the final triangulation would have at least $L_0(M)$ and at most $U_0(M) = 4e - 4n - 3b + 2k + 4(2 - \chi)$ triangles, which is better than the barycentric subdivision but worse than the output of the new algorithm described in this paper.

2 Definitions

For this paper, we define a *surface* as a connected compact topological space X where every point p has a neighborhood that is either homeomorphic to the plane $\mathbb{R}^2$, or homeomorphic to the closed half-plane $\mathbb{H}^2 = \{(x, y) : x, y \in \mathbb{R} \wedge x \geq 0\}$, with p corresponding to the origin. Points of the second type comprise the *border* of X, while the others comprise its *interior*. Note that the border must be the union of a finite number (possibly zero) of connected components, each homeomorphic to the circle $S1$. See Figure 1a. We do not require surfaces to be orientable [8].

A *topological map* M is a partition of a surface, denoted by $\mathcal{S}_M$, into a finite collection of *parts*, comprising its *vertices* $\mathcal{V}_M$, *edges* $\mathcal{E}_M$, and *faces* $\mathcal{F}_M$, such that: (i) each vertex is a singleton set; (ii) each edge is homeomorphic to $\mathbb{R}$; (iii) each face is homeomorphic to $\mathbb{R}^2$; (iv) the boundary in $\mathcal{S}_M$ of every edge is a pair of distinct vertices; (v) the boundary in $\mathcal{S}_M$ of every face is a cycle of $k \geq 2$ distinct vertices and k distinct edges. If $k = 3$ for every face, then M is said to be a *triangulation*.

With this definition, the border of $\mathcal{S}_M$ is necessarily the union of a subset of the edges and vertices of T, the *border edges* and *border vertices*. Any edge (or vertex) of M that is not on the border is an *interior edge* (or *interior vertex*). A part p of M is said to *incide on* another part q if one of them is contained in the boundary of the other. An interior edge is incident to exactly two faces of M; whereas any border edge is incident to exactly one face. These faces are called the *wings* of the edge. See Figure 1b.

Note that our definitions exclude certain partitions of the surface (see Figure 2). In particular, a map M cannot have a vertex incident to more than two border edges. Such a vertex would be a *local cut point* of the space $\mathcal{S}_M$, which we define as a point p in a topological space with a neighborhood X such that X is connected but $X \setminus p$ is not. Also, a map cannot have loops, but may have parallel edges.

A *sub-map* M' of a map M is a subset of the faces of M, together with all (and only) the edges and vertices that are incident to those faces. A sub-map has all the defining properties of a map, except that its underlying space $\mathcal{S}_{M'}$ may be disconnected, and may have local cut points. The border edges of M' are the edges of M that are incident to only

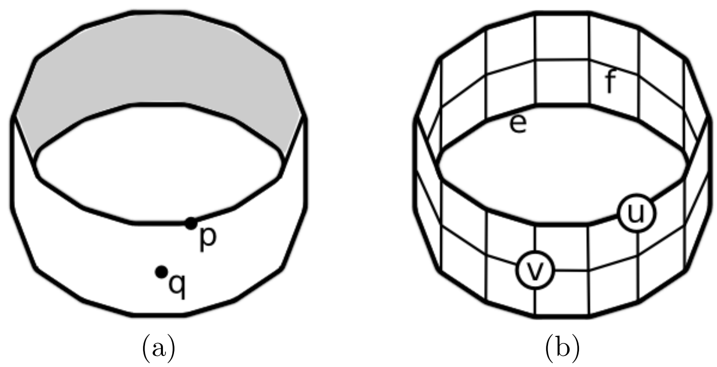


Figure 1: (a) A surface homeomorphic to the side surface of a cylinder; point p is on the border, and q is in the interior. The border has two connected components. (b) A map on that surface, with a border vertex u, an interior vertex v, a border edge e and an interior edge f.

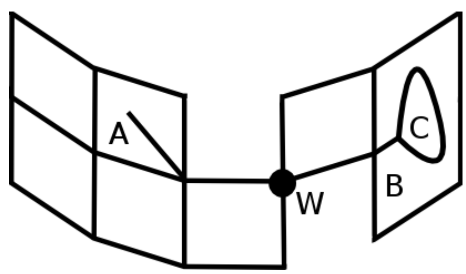


Figure 2: A partition of a topological space that is not a map. It has a local cut point w, two faces (A and B) with repeated vertices and repeated edges on their boundaries, and a face C with only one vertex and only one edge.

one face of M'. The border vertices of M' are the ones incident to those edges. Note that the local cut points of $\mathcal{S}_{M'}$ are the vertices of M' that are incident to four or more border edges of M'. Especially, if every face of M' is a triangle, then M' is a *sub-triangulation*.

A map T is said to be a *subdivision* of another map M if $\mathcal{S}_T = \mathcal{S}_M$ and every part of M is the union of some subset of the parts of T.

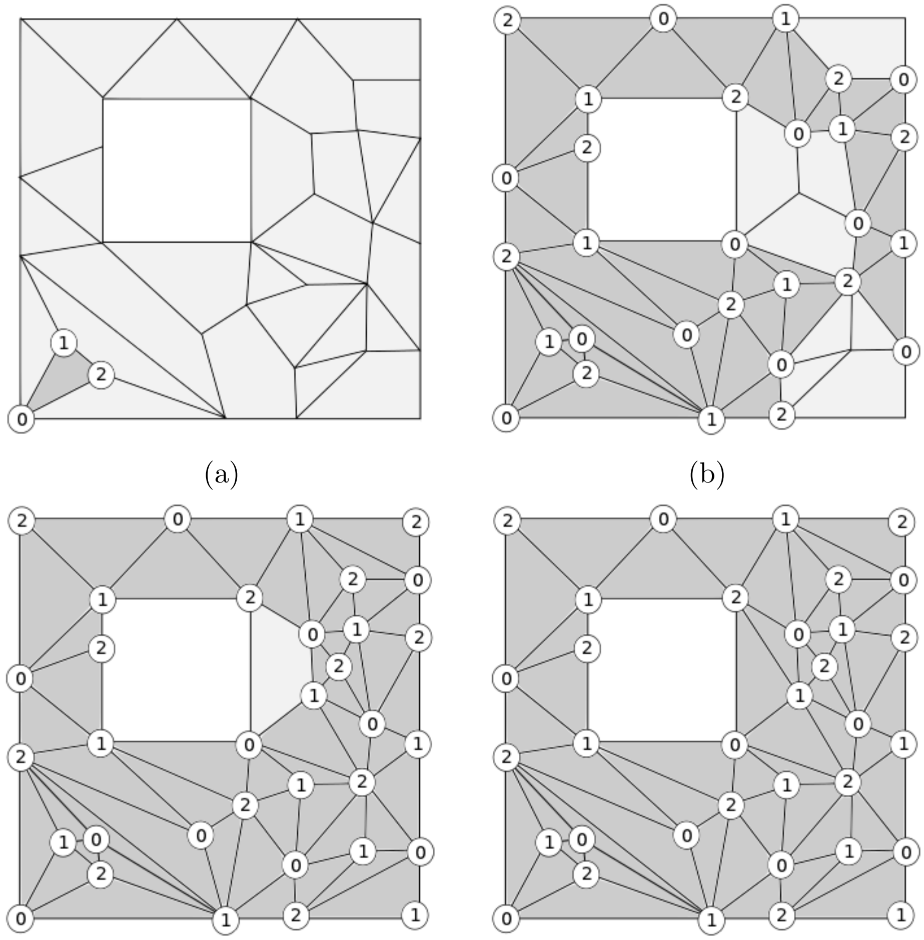
The *Euler characteristic* of a map M is the number $\chi_M = |\mathcal{V}_M| - |\mathcal{E}_M| + |\mathcal{F}_M|$. It is a topological property of the surface $\mathcal{S}_M$ alone [1], being (for example) 2 for the sphere $\mathbb{S}^2$, 0 for the torus $\mathbb{T}^2 = \mathbb{S}^1 \times \mathbb{S}^1$, and 1 for the closed unit disk $\mathbb{D}^2 = \{\ (x, y) : x^2 + y^2 \leq 1\ \}$.

3 The algorithm

3.1 Overview

Our algorithm receives a map M as input, and outputs a triangulation T, that is a subdivision of M, and a 3-coloration for T. The algorithm operates on one face of M at a time, replacing it by triangles, in such a way that the part of the triangulation that has already been processed is 3-vertex-colorable.

In the description of the algorithm, we will denote by T' the sub-triangulation of T that has already been built, and by M' the remainder of M, that still remains to be processed. Consequently, $\mathcal{S}_{M'} \cup \mathcal{S}_{T'} = \mathcal{S}_M$. Each vertex v of T' has an assigned color $\lambda(v) \in \{0, 1, 2\}$. See Figure 3 for an example where M is an uncolored triangulation.



(a) (b)

(c) (d)

Figure 3: Processing of a sample map M, showing the situation (a) just before, (b) during, and (c) just after the main loop of the 3-coloring algorithm, and (d) the final result T. The faces of T' are in dark gray and those of M' in light gray. The white square is a hole in $\mathcal{S}_M$.

While T' is a valid sub-triangulation of T (as defined in Section 2), M' is a sub-map of M, except that each edge of M in the border of $\mathcal{S}_{M'}$ may be subdivided into two edges and one vertex.

We define the *front* as the set of vertices and edges shared by both M' and T'. Every edge in the front lies between a face of M' and a face of T'. The border edges of T' or M' are either contained in the border of $\mathcal{S}_M$, or are part of the front.

The surface $\mathcal{S}_{T'}$ of T' is always connected once T' has been initialized. This follows from the fact that the algorithm only processes a face m of M' if m is incident to some edge in the front (that is already incident to a triangle of T').

3.2 The extended dual graph

Although M' may be split into two or more connected components at some stage, that may only happen, in a sense, because of the "holes" bounded by the border of $\mathcal{S}_M$. The triangulation M' would remain connected if those holes were filled in the proper way.

In order to formalize this statement (which is important for the proof of correctness) we define the *extended dual graph* M'^* that represents the connectivity of M'. The vertices of M'^* are all the faces of M', plus a *null vertex* $m_\emptyset$ if M' has any border edges of M. The edges of M'^* are all the interior edges of M', plus the border edges of M' that are border edges of M. Each interior edge e of M' connects in M'^* its two wings in M'. Each border edge of M' that is a border edge of M connects in M'^* the vertex $m_\emptyset$ to its wing in M'. See Figure 4.

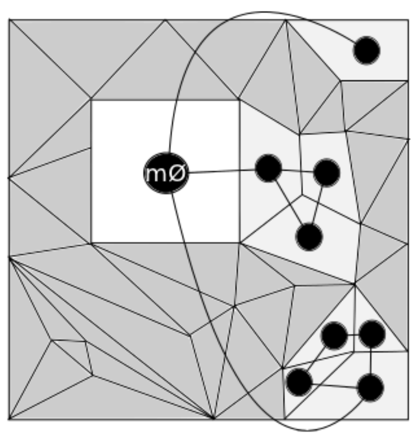


Figure 4: The extended dual graph corresponding to the situation of Figure 3(b).

Throughout the algorithm, the graph M'^* remains connected. Note that the vertex $m_\emptyset$ may provide a path between two connected components of M', as long as each still has at least one border edge of M.

3.3 Detailed algorithm

The algorithm is formally defined by the procedure *Triangulate(M)* below. It receives a topological map M as input, and outputs a topological triangulation T for M and a 3-coloration λ for the vertices of T. If the faces of M are convex polygons in some euclidean space, then the faces of T will be geometric non-overlapping triangles.

1. Let $M' \leftarrow M$ and $T' \leftarrow \{\}$. Choose any face m of M and $Process(m, T', M')$.

2. While $|\mathcal{V}_{M'^*}| \geq 2$:

   (a) find a face m of M' that is incident to at least one edge in the front, and whose removal will not disconnect the graph M'^*.

   (b) do $Process(m, T', M')$.

3. Let m be the last vertex of M'^*. If $m \neq m_\emptyset$, do $Process(m, T', M')$.

   Return $T \leftarrow T'$.

Each call of $Process(m, T', M')$ will remove the face m from M' and will add a triangulation S of m to T', in such a way that T' remains 3-vertex-colored. When we remove a face m from M' we also remove from M' any vertices and edges that are not incident to any other face of M'. In this way, M' remains a sub-map of M, except that each edge of M in the border of $\mathcal{S}_{M'}$ may be subdivided into two edges and one vertex. Also, the front remains a set of edges and vertices shared by M' and T'.

Formally, $Process(m, T', M')$ consists of four steps:

1. $Colorize(m)$

2. $M' \leftarrow M' - m$

3. $S \leftarrow Split(m)$

4. $T' \leftarrow T' \cup S$

The procedures *Colorize* and *Split* will be detailed subsequently.

3.4 The Colorize procedure

The input of the *Colorize* procedure is a face m of M'. Its effect is to assign colors to every uncolored vertices incident to m so that λ is still a valid 3-coloration.

Through the procedure, we denote by $c(0)$, $c(1)$ and $c(2)$ the number of vertices incident to m with colors 0, 1 and 2, respectively.

1. For every edge e incident to m that is not in the front, if both of its incident vertices, u and v, are in the front and $\lambda(u) = \lambda(v)$, subdivide e into two edges and a vertex w.

2. While there is some uncolored vertex u incident to m and some color i such that $c(i) = 0$, set $\lambda(u) \leftarrow i$.

3. Finally, for every uncolored vertex v incident to m, set $\lambda(v)$ to any color such that λ is a 3-coloration for m.

See Figure 5.

Note that any vertex colored by the *Colorize* procedure is not in the front and therefore it is not a neighbor, in T', of an other vertex of T'. Thus, in step 3, as long as λ is a 3-coloration for m, it will also be a 3-coloration for $m \cup T'$.

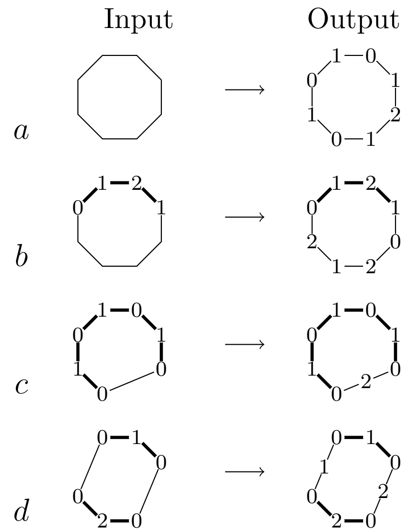


Figure 5: Examples of the *Colorize* procedure. The thick lines represent edges in the front.

3.5 The Split procedure

The *Split* procedure is called after the *Colorize* procedure. Its input is a face m such that every two consecutive vertices along the border of m are colored with different colors.

During this procedure, the face m will progressively shrink and eventually disappear. Meanwhile, the procedure will build a triangulation S that will cover the original face m and its incident vertices and edges. The result of *Split* is the 3-colored triangulation S.

As in *Colorize*, through the procedure, we denote by $c(0)$, $c(1)$ and $c(2)$ the number of vertices in m with colors 0, 1 and 2, respectively.

1. If $c(i) = 0$ for any $0 \leq i \leq 2$, create a vertex u inside m. Set $\lambda(u) \leftarrow i$. Create an edge between u and every vertex of m but u. Add the resulting triangles to S. Return S.

2. Otherwise, while $c(0) \geq 2$ and $c(1) \geq 2$ and $c(2) \geq 2$ do:

   There must be a vertex v, in m, with neighbors u, w such that $\lambda(v)$, $\lambda(u)$ and $\lambda(w)$ have three different colors. Add an edge between u and w, splitting m into a face m' and a triangle t. Add t into S and set $m \leftarrow m'$.

3. There must be a color i and a vertex u of m such that $\lambda(u) = i$ and $c(i) = 1$. Add an edge between u and every vertex of m not adjacent to u. Add the resulting triangles to S. Return S.

Figure 6 shows three examples for the *Split* procedure.

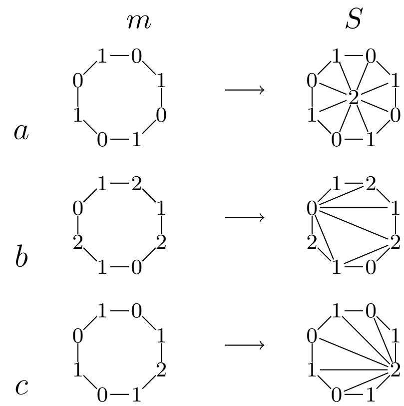


Figure 6: Examples of the *Split* procedure, when the condition of the step 1 is satisfied (a), when the condition of the step 2 is satisfied at least once (b), and when only step 3 is executed (c).

3.6 Selecting the next face

Let X_m be, for every face m of M', the set of edges and vertices incident to m that are not in the front. For efficient implementation of step 2a of *Triangulate* we maintain for each m a counter N_m with the number of connected components of X_m. Every time a vertex v is colored, N_m is incremented for every face $m \in M'$ incident to v. Every time an edge e is added to the front, N_m is decremented for the face $m \in M'$ incident to e.

We also maintain a priority queue with all the faces m of M' incident to at least one edge in the front. The queue has only two priorities: if $N_m = 1$, m has high priority, otherwise m has low priority. In step 2a, the algorithm will only select a face with low priority if no face with high priority remains in the queue. The removal of any face with high priority from M' does not disconnect M'^*, as we will see in the next section. If there are only low priority faces left, all faces m whose removal from M' does not disconnect M'^* are found by the depth-first search algorithm of Hopcroft and Tarjan [4]. As we will see in the next section, at least one of them is incident to the front and will be selected.

4 Correctness

Throughout the execution of the algorithm, every time a face m is processed, the union of the parts that are removed from M', including m itself and some of its incident vertices and edges, will be added to T' if they are not there yet. The edges and vertices incident to m that are not removed from M' will also be added to T', updating the front.

The result S of the *Split* procedure is always a triangulation and $S \cup T'$ is sub-triangulation of T. Except for step 1 of *Triangulate*, m is incident to a border edge of T' and therefore $S \cup T'$ will be connected (including just after step 1 as $T' = S$).

Any new vertex is either added to an edge of M' at step 1 of the *Colorize* procedure or inside m at step 1 of the *Split* procedure. Anyway, the result of these procedures are subdivisions of the original face.

Finally, the colors assigned to vertices incident to m, in steps 2 and 3 of the *Colorize* procedure or step 1 of the *Split* procedure, extend λ to a valid 3-coloration of $S \cup T'$.

Thus, after each face is processed, the properties described in the algorithm overview can be verified:

- the union of the parts of T' and M' is the surface $\mathcal{S}_M$;
- the intersection of $\mathcal{S}_{T'}$ and $\mathcal{S}_{M'}$ (the front) consists only of edges and vertices of T' and M';
- T' is a valid connected sub-triangulation of T;
- M' is a valid sub-map of M, except that each edge of M in the border of $\mathcal{S}_{M'}$ may be subdivided into two edges and one vertex;
- T' is a valid subdivision of a sub-map of M;
- λ is a valid 3-coloring of T'.

Moreover, after the execution of step 3 of *Triangulate*, M' will have no faces and T' will be a valid subdivision of M. Therefore, T will be a valid subdivision of M and λ will be a valid 3-coloring of T. Note that the algorithm terminates, because at each iteration the map M' loses one face.

To prove the correctness of the algorithm it remains to be shown that there is always a face m that satisfies the conditions of step 2a of *Triangulate*. Lemmas 1 and 2 and Corollary 1 guarantee this.

Given the implementation of step 2a detailed in Section 3.6 we must also prove that, for any face m of M incident to at least one edge in the front, if $N_m = 1$ then the removal of m from M' does not disconnect M'^*. Lemma 3 guarantees this.

Lemma 1. *Before every execution of step 2a of Triangulate the front has at least one edge.*

Proof. Just before processing face m, at least one edge e incident to it is not in the border of T'. In step 1 of *Triangulate* e can be any edge; in step 2b, the existence of such an edge follows from the condition $|\mathcal{V}_{M'^*}| \geq 2$. Moreover, e will be in the border of T' after the execution of these steps, by construction of the *Split* procedure. Therefore, every execution of the steps 1 and 2b creates at least one new edge in the border of T'. Thus, before each iteration of step 2a the border of T' is not empty.

Note that $\mathcal{F}_{M'}$ is not empty, because the main loop of the algorithm processes one face of M' at a time and stops if M'^* is only $\{m_\emptyset\}$ or if $\mathcal{F}_{M'}$ has only one face (in the case that $m_\emptyset$ doesn't exist). Since $\mathcal{S}_M$ is connected, the two closed subsets $\mathcal{S}_{T'}$ and $\mathcal{S}_{M'}$ of $\mathcal{S}_M$ must have a non-empty intersection. Since $\mathcal{F}_{T'}$ and $\mathcal{F}_{M'}$ are disjoint and $\mathcal{S}_M$ has no local cut points, the intersection must include at least one whole edge of M', which is also an edge of T'. By definition, this edge must be part of the front. $\square$

For the next lemma we define $\mathcal{V}^B_{M'^*}$ as the set of vertices of M'^* whose dual faces are incident to at least one edge in the front.

Lemma 2. *Let u^* be a vertex of M'^* and G be a connected component of $M'^* - u^*$. If G has at least one vertex in $\mathcal{V}^B_{M'^*}$ and doesn't have the vertex $m_\emptyset$, then there must be a vertex in both G and $\mathcal{V}^B_{M'^*}$ whose removal does not disconnect the induced subgraph $H = M'^*[\mathcal{V}_G \cup u^*]$.*

Proof. The graph H must have at least two vertices (u^* and a vertex in both G and $\mathcal{V}^B_{M'^*}$). H must also be connected. If H has exactly two vertices, u^* and $v^* \in \mathcal{V}^B_{M'^*}$ there is an edge connecting them. The removal of v^* doesn't disconnect H and the lemma is true.

Suppose by induction that the lemma is true for any k such that $2 \leq |\mathcal{V}_H| < k$. We want to prove the lemma for $|\mathcal{V}_H| = k$.

Chose a vertex v^* in both G and $\mathcal{V}^B_{M'^*}$. If the removal of v^* from G does not disconnect H, we are done. Otherwise H is the sum of two or more connected subgraphs $H_1, H_2, \ldots, H_n$, $n \geq 2$ such that $|\mathcal{V}_{H_i}| \geq 2$ for any $1 \leq i \leq n$, $H_i \cap H_j = v^*$ for any $1 \leq i < j \leq n$, $u^* \in \mathcal{V}_{H_n}$, and $\bigcup_i^n H_i = H$. Note that $u^* \notin \mathcal{V}_{H_1}$. Let G_1 be $H_1 - v^*$.

We will show now that $\mathcal{V}_{G_1} \cap \mathcal{V}^B_{M'^*} \neq \emptyset$.

The graph G_1 must have a vertex w^* that is adjacent to v^* in H_1. Let w and v be the dual faces of w^* and v^* in M'. The set of edges E_{wv} incident to both w and v must then be part of the boundary B of the sub-map represented by G_1 in M'. Also, B does not contain any border edges of M because $m_\emptyset$ is either a connected component of $M'^* - u^*$ distinct from G or $m_\emptyset$ is u^* itself. In either case, no vertices of G_1 are incident $m_\emptyset$. Moreover, B does not contain any interior edges of M' except those of E_{wv}, since every vertex of M'^* that is adjacent to a vertex of G_1, except v^*, is also in G_1. Therefore, every edge of B is either incident to the face v or is an edge in the front. On the other hand, v is bounded by a cycle of distinct vertices and distinct edges. That cycle contains at least one edge in the front. Thus, the edges and vertices incident to v that are not in the front do not form any cycle and therefore they cannot comprise all vertices and edges of B. It follows that B must contain at least one edge in the front incident to a face f different from v.

Note that $|\mathcal{V}_{G_1}| < |\mathcal{V}_G|$, $m_\emptyset$ is not part of G_1, G_1 has at least one vertex f^* in $\mathcal{V}^B_{M'^*}$ and G_1 is a connected component of $M'^* - v^*$. By induction hypothesis, there is a vertex from $\mathcal{V}^B_{M'^*}$ whose removal does not disconnect H_1. Consequently, its removal does not disconnect H.

□

Corollary 1. *Just before every execution of step 2a of Triangulate, there is a face m of M' that has an edge e in the front, whose removal from M' does not disconnect the M'^* graph.*

Proof. $\mathcal{V}^B_{M'^*}$ cannot be empty due to Lemma 1. If $m_\emptyset$ exists, let u^* be $m_\emptyset$. Otherwise, if there is a vertex of M'^* that is not in $\mathcal{V}^B_{M'^*}$, let u^* be that vertex. Otherwise, let u^* be any vertex of $\mathcal{V}^B_{M'^*}$. Since $|\mathcal{V}_{M'^*}| \geq 2$, in all cases, there is a graph G that is a connected component of $M'^* - u^*$ and that contains a vertex from $\mathcal{V}^B_{M'^*}$. G does not contain $m_\emptyset$. By Lemma 2, there is a vertex in both $\mathcal{V}^B_{M'^*}$ and G whose removal does not disconnect the induced subgraph $H = M'^*[\mathcal{V}_G \cup u^*]$.

If $M'^* - u^*$ has only one connected component, $H = M'^*$ and the corollary is true. Otherwise, M'^* is the sum of two or more connected subgraphs with u^*, and only u^*, in common. One of these subgraphs is H. There is a path from any vertex of M'^* outside

H to a vertex in H passing through u^*. Therefore, M'^* is not disconnected by removing a vertex that does not disconnect H and the corollary is true.

□

Lemma 3. *The removal of any face m with $N_m = 1$ from M' will never disconnect the M'^* graph.*

Proof. Since $N_m = 1$ there is an alternating path $\{e_0, v_1, e_1, v_2, e_2, \ldots, v_n, e_n\}$ in M' that comprises all edges and vertices incident to m that are not in the front. For each vertex v_i, $1 \leq i \leq n$, there is a circular alternating path in M'^* $C_i = \{m^*, e^*_{i1}, v^*_{i1}, e^*_{i2}, v^*_{i2}, \ldots, v^*_{i(k_i-1)}, e^*_{ik_i}, m^*\}$ of vertices and edges whose primal in M' are the faces and edges incident to v_i. The vertex m^* is the dual of m, the vertex v^*_{i1} is the dual of the face incident to e_{i-1} and the vertex $v^*_{i(k_i-1)}$ is the dual of the face incident to e_i. The path C_i is a cycle because all faces incident to v_i are faces of M' (note that $m_\emptyset$ is part of C_i if and only if v_i is a border vertex).

For every $1 \leq i \leq n-1$ note that $v^*_{i(k_i-1)} = v^*_{(i+1)1}$. Therefore, inductively one can see that there is a path P from v_{11} to $v_{n(k^n-1)}$ that comprises all vertices and edges of all the paths $C_1, C_2, \ldots, C_n$ except m^*. Therefore the removal of m^* does not disconnect P. Since all vertices adjacent to m^* are vertices of P, the removal of m^* does not disconnect M'^* either. □

From these lemmas we conclude the following:

Theorem 1. *For any triangulation M, Triangulate(M) returns a 3-colored subdivision T of M.*

5 Efficiency

5.1 Output upper bounds

We will prove two upper bounds $U_1(M)$ and $U_2(M)$ for the size $|\mathcal{F}_T|$ of the output T of *Triangulate (M)*. Bound $U_1(M)$ is defined for any map M as:

$$U_1(M) = 2(2-\chi) + \sum_{m \in \mathcal{F}_M} d(m)$$

where $d(m)$ is the degree of the face m. If the average degree of the faces is 4 or more, then $U_1(M)$ is less than the upper bound $U_0(M)$ for the naive algorithm described in Section 1. Note that $\sum_{m \in \mathcal{F}_M} d(m) = 2|\mathcal{E}_M| - |\mathcal{E}^B_M|$, where $\mathcal{E}^B_M$ is the set of border edges of M.

Bound $U_2(M)$ is defined only if all faces of M have the same degree D:

$$U_2(M) = (D-1)|\mathcal{F}_M| + 4(2-\chi) - |\mathcal{E}^B_M|$$

Note that $U_2(M) < U_1(M)$. If $D \geq 4$, then $U_2(M) < U_0$. If $D = 3$, then $U_0(M) = U_2(M) = 2|\mathcal{F}_M| + 4(2-\chi) - |\mathcal{E}^B_M|$, which is the same upper bound given in [3].

For the following lemmas we define $\mathcal{K}$ as the number of faces of M' that will have all its incident vertices already colored when the *Colorize* procedure is called.

Lemma 4 and Theorem 2 guarantee that $U_1(M)$ is an upper bound for $|\mathcal{F}_T|$.

Lemma 4. *For any map M, Triangulate(M) returns a 3-colored triangulation T of M with at most $|\mathcal{V}_M| + \mathcal{K} + 2 - \chi$ vertices.*

Proof. Let's consider the implementation of step 2a of *Triangulate* described in Section 3.6. Also, we consider $m = m'$ and $m = m''$ for steps 1 and 3, respectively.

Note that all vertices of M are also vertices of T, so $|\mathcal{V}_T|$ will be $|\mathcal{V}_M|$ plus any new vertices created.

Note also that step 1 of *Triangulate* never creates a new vertex and step 3 creates at most one. Finally note that when *Colorize*(m'') is Called, all vertices incident to m'' are already colored.

By inspection of the *Colorize* procedure, one can see that for any m selected by step 2a of *Triangulate*, at most N_m new vertices are created. Moreover, if m has at least one uncolored vertex when the *Colorize* procedure is called, at most $N_m - 1$ vertices are created. By inspection of both *Colorize* and *Split* procedures, one can see that for any m as input, if any vertex (either of M or a new one) is colored in *Colorize* procedure, no vertex will be crated in *Split* procedure. Therefore $|\mathcal{V}_T| \leq |\mathcal{V}_M| + \mathcal{K} + \sum_{m \in \{\mathcal{F}_M - m'' - m'\}} (N_m - 1)$.

For each iteration of step 2a, since M'^* is connected, $X_m \neq \emptyset$, and since m has at least one edge in the front, X_m does not comprise the whole boundary of m. Each component of X_m has, then, $n \geq 1$ edges and $n - 1$ vertices and therefore, $|\mathcal{E}_{X_m}|$ - $|\mathcal{V}_{X_m}| = N_m$. For step 1, $X_{m'}$ comprises the whole boundary of m' and $|\mathcal{E}_{X_{m'}}|$ - $|\mathcal{V}_{X_{m'}}| = 0$. For step 3, if m'' exists, $X_{m''} \neq \emptyset$ and $|\mathcal{E}_{X_{m''}}|$ - $|\mathcal{V}_{X_{m''}}| = 0$.

Let $\chi_{T'}$ be the Euler characteristic of T'. Note that by removing m from M' and adding it in T', without triangulating m, we have the same effect on the value of $\chi_{T'}$ as removing m from M' and adding the triangulation S in T', as noted by *Process*. Therefore, just after m is processed, $\chi_{T'}$ will be increased or decreased, according to the Euler's polyhedron formula, by $|\mathcal{F}_m| + |\mathcal{V}_X| - |\mathcal{E}_X|$, which is equal to 1 for steps 1 and 3 (if m'' exists) or $1 - N_m$ for step 2a. At the start of the execution of the algorithm $\chi_{T'} = 0$ and at the end, $\chi_{T'} = \chi$. We conclude that $\chi \leq 2 + \sum_{m \in \{\mathcal{F}_M - m'' - m'\}} (1 - N_m)$. But we also know that $|\mathcal{V}_T| \leq |\mathcal{V}_M| + \mathcal{K} + \sum_{m \in \{\mathcal{F}_M - m'' - m'\}} (N_m - 1)$.

With both equations we prove that $|\mathcal{V}_T| \leq |\mathcal{V}_M| + \mathcal{K} + 2 - \chi$.

□

Theorem 2. *For any map M, Triangulate(M) returns a 3-colored triangulation T of M with at most $2(2 - \chi) + \sum_{m \in \mathcal{F}_M} d(m)$ faces.*

Proof. By Euler's formula $|\mathcal{V}_T| + |\mathcal{F}_T| - |\mathcal{E}_T| = |\mathcal{V}_M| + |\mathcal{F}_M| - |\mathcal{E}_M|$. Since each edge contributes twice for the degree of incident faces or incident holes along the border of the map,

$$|\mathcal{V}_T| + |\mathcal{F}_T| - \frac{(3|\mathcal{F}_T| + |\mathcal{E}_T^B|)}{2} = |\mathcal{V}_M| + |\mathcal{F}_M| - \frac{(\sum_{m \in \mathcal{F}_M} d(m) + |\mathcal{E}_M^B|)}{2}$$

We have that $|\mathcal{E}_T^B| \geq |\mathcal{E}_M^B|$ and, by Lemma 4, $|\mathcal{V}_T| \leq |\mathcal{V}_M| + \mathcal{K} + 2 - \chi$. Thus:

$$|\mathcal{F}_T| \leq 2\mathcal{K} + 2(2 - \chi) + \sum_{m \in \mathcal{F}_M} d(m) - 2|\mathcal{F}_M|$$

$\mathcal{K}$ cannot be more than $|\mathcal{F}_M|$ itself and the lemma is proved.

□

Lemma 5 and Theorem 3 guarantee that $U_2(M)$ is an upper bound for $|\mathcal{F}_T|$ if all faces of M have the same degree.

Lemma 5. *If all faces of the input map M have the same degree D,* $2\mathcal{K} \leq |\mathcal{F}_M| + 2(2-\chi) - |\mathcal{E}_M^B|$.

Proof. The first face of M' processed by the algorithm will require the coloring of all its incident vertices, which are also vertices of M. During the algorithm, $\mathcal{K}$ faces of M' will not require any vertex coloring. By the conditions of step 2a of *Triangulate*, every processed face will require the coloring of at most $D-2$ vertices of M. Therefore, by counting the number of vertices of M that will be colored, face by face, we have:

$$|\mathcal{V}_M| \leq D + (|\mathcal{F}_M| - \mathcal{K} - 1)(D-2)$$

Each edge contributes twice for the degree of incident faces or incident holes along the border of the map. Also, $|\mathcal{V}_M| = |\mathcal{E}_M| - |\mathcal{F}_M| + \chi$. Furthermore, $|\mathcal{E}_M| = D(|\mathcal{F}_M| + |\mathcal{E}_M^B|)/2$ Thus:

$$\frac{D(|\mathcal{F}_M| + |\mathcal{E}_M^B|)}{2} - |\mathcal{F}_M| + \chi \leq (D-2)(|\mathcal{F}_M| - \mathcal{K}) + 2$$

After some algebraic manipulation:

$$2\mathcal{K} \leq |\mathcal{F}_M| + 2(2-\chi) - |\mathcal{E}_M^B|$$

and the lemma is proved.

□

Theorem 3. *If all faces of the input map M have the same degree D, Triangulate(M) returns a 3-colored triangulation T of M with at most $(D-1)|\mathcal{F}_M| + 4(2-\chi) - |\mathcal{E}_M^B|$ faces.*

Proof. By Theorem 2:

$$|\mathcal{F}_T| \leq 2\mathcal{K} + 2(2-\chi) + \sum_{m \in \mathcal{F}_M} d(m) - 2|\mathcal{F}_M|$$

Replacing $2\mathcal{K}$ according to Lemma 5 and replacing $\sum_{m \in \mathcal{F}_M} d(m)$ by $D|\mathcal{F}_M|$, the theorem is proved.

□

The upper bounds U_1 and U_2 are not always achieved. The average $|\mathcal{F}_T|$ can be much less than these bounds, especially for maps with large average face degree. See Section 7 for more details.

5.2 Output lower bound for triangulations as input

Note that a triangulation is 3-colorable only if all interior vertices have even degree. This condition is in fact sufficient for planar triangulations [9]. Therefore, if all faces of M are triangles then, for each odd interior vertex, at least one edge incident to that vertex must be created during the execution of the algorithm. It follows that at least one additional triangle must be present in T (compared to M) for each odd interior vertex. Thus, a lower bound for $|\mathcal{F}_T|$ is $L_1(M) = |\mathcal{F}_M| + |\mathcal{V}_M^o|$, where $\mathcal{V}_M^o$ is the set of odd interior vertices of T.

5.3 Time complexity

The algorithm can be executed in $O((3-\chi)|\mathcal{E}_M|)$ time. Therefore, for surfaces with the same value of χ the algorithm runs in linear time with respect to $|\mathcal{E}_M|$.

Lemma 6 gives a bound for the execution of all steps of *Triangulate*, except for step 2a.

Lemma 6. *The execution of all iterations of step 2b in addition to the execution of steps 1 and 3 of Triangulate can be done in $O(|\mathcal{E}_M|)$ time.*

Proof. Steps 1, 3 and every iteration of step 2b of *Triangulate* process exactly one face m of M. At step 1, m can be any face of M and, at step 3, if m exists, it is the only face of M'. Therefore, the selection of m takes $O(1)$ in these cases. At step 2b, m is the face previously selected by step 2a.

To process the face m the procedures *Colorize* and *Split* are called once each. If we prove that every edge incident to m is accessed a constant amount of time by both *Colorize* and *Split* procedures, since every edge of M is incident to one or two faces, the time complexity to process all faces of M is $O(|\mathcal{E}_M|)$.

One can see that each step of the *Colorize* procedure can be computed by verifying every edge (and its incident vertices) incident to m once.

Each of the steps 1 and 3 of the *Split* procedure accesses every edge incident to m once.

The first iteration of step 2 of the *Split* procedure can identify all possible sets of three consecutive vertices with three different colors by verifying every edge (and its incident vertices) incident to m twice. Every time a triangle is created in step 2 of the *Split* procedure, a constant number of these sets are updated (those sets that contain some of the vertices incident to the triangle just created), each one in constant time.

Therefore both *Colorize* and *Split* procedures take $O(|\mathcal{E}_M|)$ time to process all faces and the lemma is true.

□

Now, we must show that the time complexity for all iterations of step 2a of *Triangulate* is $O((3-\chi)|\mathcal{E}_M|)$.

To achieve this goal consider the implementation defined in Section 3.6. Note that each face m will update N_m twice for each incident edge. Therefore the update of N_m for all $m \in M$ can be done in $O(|\mathcal{E}_M|)$. The priority queue is updated in constant time.

Any face $m \in M$ with $N_m = 1$ that is incident to at least one edge in the front is found in $O(1)$ and its removal does not disconnect the M'^* graph. Therefore all these faces together are found in $\mathrm{O}(\mathcal{E}_M)$.

Since every iteration of *Triangulate* after step 1 $N_m \geq 1$, it remains to prove Lemma 7 to complete the proof of time complexity.

Lemma 7. *At most $2-\chi$ iterations of Triangulate will process a face m with $N_m > 1$ that will not disconnect the M'^* graph. All these faces together can be found in $O((2-\chi)|\mathcal{E}_M|)$ time.*

Proof. Let $\chi_{T'}$ be the Euler Characteristic of T' and let m' and m'' be the faces processed by steps 1 and 3 of *Triangulate*. Processing m' increases $\chi_{T'}$ by one. If m'' exists, processing it also increases $\chi_{T'}$ by one. On the other hand, one can see that every iteration of step 2b will decrease $\chi_{T'}$ by $1 - N_m$. Since every face processed is incident to the front, $N_m \geq 1$ and $\chi_{T'}$ will never increase on any iteration of step 2b. At the start of the execution of algorithm, $\chi_{T'} = 0$ and at the end $\chi_{T'} = \chi$. Therefore, $\chi \leq 2 - \sum_{m \in \{\mathcal{F}_M \setminus m'' \setminus m'\}} (N_m - 1)$. The number of processed faces with $N_m > 1$ will be no more than the value of the later sum and thus at most $2-\chi$.

If every face m incident to the front has $N_m > 1$, then all faces whose removal from M' does not disconnect M'^* are found by the depth-first search algorithm of Hopcroft and Tarjan [4] in $O(|\mathcal{E}_{M'^*}|)$ time. By Lemma 2, at least one of these faces is incident to at least one edge in the front. Since $|\mathcal{E}_{M'^*}| = |\mathcal{E}_{M'}|$, each face of m with $N_m > 1$ that does not disconnect M'^* can be found in $O(|\mathcal{E}_{M'}|)$. Therefore $2-\chi$ of these faces can be found in $O((2-\chi)|\mathcal{E}_M|)$ time.

□

Adding the execution of all steps of *Triangulate* we prove Theorem 4.

Theorem 4. *The algorithm can be executed in $O((3-\chi)|\mathcal{E}_M|)$ time.*

6 Efficient Face Selection Heuristic

Section 5 defines $\mathcal{K}$ as the number of faces of M' that will have all its incident vertices already colored when the *Colorize* procedure is called. Lemma 4 proves that at most $\mathcal{K}+2-\chi$ new vertices are created by the algorithm. The value of $\mathcal{K}$ is not a property of the map and can vary significantly depending on the order that the faces are processed. Figure 7 shows two orders of processing for the same input map. In the first example (top) $\mathcal{K}$ is almost equal to $\mathcal{F}_M$; and moreover, a new vertex is created for each face that counts for $\mathcal{K}$. Consequently, the size of the output is close to the upper bound. In the second example (bottom) $\mathcal{K}$ is zero and the algorithm creates no new vertices, so the number of triangles of the output will be the same as in a minimal uncolored triangulation of the map.

In this section we describe an heuristic for selecting the next face to be processed by the algorithm that tries to minimize $\mathcal{K}$ in the hope of minimizing the number of triangles of the output.

As described in Section 3.6, the algorithm maintains a priority queue with all the faces m of M' incident to at least one edge in the front. To implement the heuristic we modify the queue so it has D priority levels instead of two, where D is the degree of the face of M with the largest degree. We also maintain, for each face m in the queue, a counter

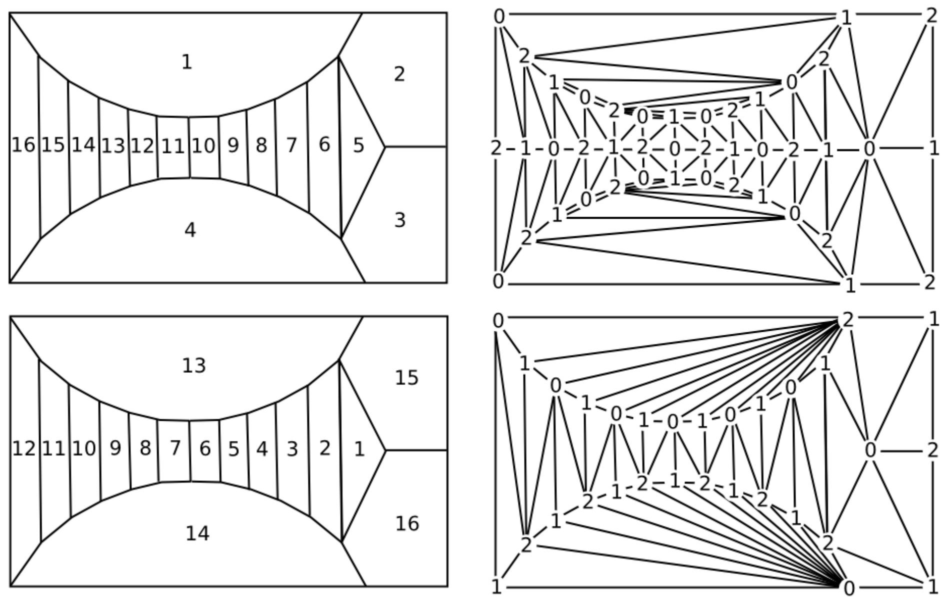


Figure 7: Effect of the processing order on the efficiency of the algorithm. The faces selected to be processed are numbered in increasing order in the input map at left. The output triangulation is at right.

$U_m = |\mathcal{V}_{X_m}|$ (the number of vertices of m not in the front). If a face m has $N_m = 1$, its priority is $D - U_m$, otherwise its priority is 1. For instance, if $N_m = 1$ and all vertices of m are colored ($U_m = 0$), m has the highest priority (D).

In step 2a of *Triangulate*, the modified algorithm still selects one of the faces in the queue with the highest priority. If that priority is 1, then $N_m > 1$ and the algorithm must search for a face in the queue that does not disconnect M'^*, as usual. Otherwise the algorithm selects any face m with that priority (which has $N_m = 1$ and therefore does not disconnect M'^*).

Insertion or deletion of a face in the queue still takes constant time. Finding a face m with highest priority in the queue takes $O(d(m))$, and therefore finding all faces with highest priority throughout the execution of the algorithm takes $O(|\mathcal{E}_M|)$ time. As before, the update of N_m for all $m \in M$ can be done in $O(|\mathcal{E}_M|)$. Also, U_m is decremented every time N_m is incremented, except when a new vertex is created (and colored) during the *Colorize* procedure. Therefore the updating of U_m does not increase the time complexity of the the algorithm, that still executes in $O((3-\chi)|\mathcal{E}_M|)$ time.

6.1 Effectiveness for triangulations

In the particular case when M is a triangulation, then every face m in the queue will have at most one uncolored vertex. Of the $|\mathcal{F}_M| - 1$ faces that are removed from the queue in steps 2b and 3 of *Triangulate*, $|\mathcal{F}_M| - 1 - \mathcal{K}$ of them will result in one additional vertex of M being colored. Since three vertices of M are colored in step 1, then $|\mathcal{V}_M| - 3 = |\mathcal{F}_M| - 1 - \mathcal{K}$, which means that $\mathcal{K} = |\mathcal{F}_M| - |\mathcal{V}_M| + 2$ and it doesn't depend on the choice of m in step 2b. Therefore the heuristic turns out to be ineffective in this particular case. This observation holds even if M has only faces of degree 2 or 3.

7 Experimental results

This section shows the result of some experiments. In all these experiments the average size of the output $|\mathcal{F}_T|$ was found to be lower than the upper bound given in Section 5, especially for maps with faces with larger average degree.

In each experiment a number of random instances were generated from a specific family of planar maps ($\chi = 1$). The size of each map was determined by a parameter n set to be approximately equal to $|\mathcal{F}_M|$. The values of n used by all experiments were 256, 512, 1024 and 2048. For each value of n, we generated 100 maps and triangulated them by the algorithm. We also computed the average values $\overline{|\mathcal{F}_M|}$ and $\overline{|\mathcal{F}_T|}$ of the input and output sizes, as well the average values $\overline{L}$ and $\overline{U}$ of some lower bound $L(M)$ and some upper bound $U(M)$ for $|\mathcal{F}_T|$.

We define *mean efficiency* of the algorithm by the formula: $\eta = (\overline{|\mathcal{F}_T|} - \overline{L})/(\overline{U} - \overline{L})$. The value of η ranges from 0 to 1; the closer η is to zero, the better. In particular, if $\eta = 0$ then $\overline{|\mathcal{F}_T|}$ was equal to the lower bound $L(M)$ for all instances and therefore the results were optimal. If $\eta = 1$ then $|\mathcal{F}_T|$ was equal to the upper bound $U(M)$ for all instances. Note that the bounds are not tight, and therefore an optimum algorithm could have $\eta > 0$, and it may be impossible for η to be exactly 1 for any family of maps.

7.1 Delaunay Triangulations

In the first experiment, each input map was a Delaunay triangulation of a set of $(n+2)/2$ random sites with a two-dimensional normal distribution. In this experiment the parameter n turns out to be $|\mathcal{F}_M| + |\mathcal{E}_M^B|$. We used for $U(M)$ the upper bound $U_2(M)$ of Section 5.1, that for this experiment turns out to be $2|\mathcal{F}_M| - |\mathcal{E}_M^B| + 4$. For the lower bound $L(M)$ we chose the formula of Section 5.2, namely $L_1(M) = |\mathcal{F}_M| + |\mathcal{V}_M^o|$. The face selection heuristic described in Section 6 was not used, since, as observed in Section 6.1, it is effective only for maps with at least some faces of degree higher than 3.

See the results in Table 1. Note that the average output size $\overline{|\mathcal{F}_T|}$ was approximately $1.5\overline{|\mathcal{F}_M|}$ and about 1/3 of the way from the lower bound to the upper bound, for all input sizes.

Table 1: Experimental results of the algorithm for Delaunay triangulations as input maps.

n	$\overline{\lvert\mathcal{F}_M\rvert}$	$\overline{\lvert\mathcal{F}_T\rvert}$	$\overline{\lvert\mathcal{F}_T\rvert}/\overline{\lvert\mathcal{F}_M\rvert}$	$\overline{L}$	$\overline{U}$	η
256	246.42	368.51	1.495	306.64	487.26	0.3425
512	501.82	750.61	1.496	624.95	997.46	0.3373
1024	1012.88	1520.94	1.502	1263.37	2018.63	0.3410
2048	2035.68	3064.35	1.505	2541.16	4063.04	0.3438

7.2 Voronoi Diagrams

In the second experiment, each input map was a Voronoi diagram of a set of n random sites with a two-dimensional normal distribution. The unbounded faces of the Voronoi diagrams were not processed. In this experiment the parameter n turns out to be $|\mathcal{F}_M|$ plus the number of unbounded faces. We used for $U(M)$ the upper bound $U_1(M)$ of Section 5.1, that for this experiment turns out to be $2|\mathcal{E}_M| - |\mathcal{E}_M^B| + 2$. We chose the lower bound $L(M)$ as the number of triangles of minimal uncolored triangulation for M, that is $L_0(M) = 2|\mathcal{E}_M| - 2|\mathcal{F}_M| - |\mathcal{E}_M^B|$.

We executed the algorithm with and without the heuristic described in Section 6 to show its importance. See the results in Tables 2 and 3. Note that without the heuristic the average output size $\overline{|\mathcal{F}_T|}$ was about 3% higher than the average size of the minimal uncolored triangulation $\overline{L}$, and about 12 times nearer to $L(M)$ than to $U(M)$ for, all input sizes. However, the results with the heuristic were even better and $\overline{|\mathcal{F}_T|}$ was between 0.01% and 0.04% higher than $\overline{L}$.

Table 2: Experimental results of the algorithm for Voronoi diagrams as input maps, without the face selection heuristic.

n	$\overline{\|\mathcal{F}_M\|}$	$\overline{\|\mathcal{F}_T\|}$	$\overline{\|\mathcal{F}_T\|}/\overline{L}$	$\overline{L}$	$\overline{U}$	η
256	245.78	1235.87	1.030	1199.35	1651.76	0.0807
512	501.09	2543.50	1.031	2466.12	3423.91	0.0808
1024	1012.03	5162.41	1.032	5004.71	6978.78	0.0799
2048	2035.05	10403.97	1.031	10091.63	14105.03	0.0778

Table 3: Experimental results of the algorithm for Voronoi diagrams as input maps, including the face selection heuristic.

n	$\overline{\|\mathcal{F}_M\|}$	$\overline{\|\mathcal{F}_T\|}$	$\overline{\|\mathcal{F}_T\|}/\overline{L}$	$\overline{L}$	$\overline{U}$	η
256	245.59	1204.29	1.3 x 10^{-4}	1204.13	1655.84	0.000354
512	500.22	2473.10	10^{-4}	2472.84	3426.61	0.000273
1024	1012.19	5030.98	1.8 x 10^{-4}	5030.06	7004.72	0.000466
2048	2035.62	10146.74	4.1 x 10^{-4}	10142.62	14158.14	0.001026

References

[1] P. S. Alexandroff. *Combinatorial Topology.* Dover Books, 1998.

[2] R. H. Bing. *Geometric Topology of 3-Manifolds*, volume 40. Colloquium publications, 1983.

[3] Lucas Moutinho Bueno and Jorge Stolfi. Economic 3-colored subdivision of triangulations. In *European Symposium on Algorithms ESA*, volume 8125, pages 265–276. Springer Berlin Heidelberg, 2013.

[4] John Hopcroft and Robert Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, 1973.

[5] Sóstenes Lins and Arnaldo Mandel. Graph-encoded 3-manifolds. *Discrete Mathematics*, 57(3):261–284, 1985.

[6] Arnaldo Jovanini Montagner. A estrutura de dados gema para representação de mapas n-dimensionais. Master's thesis, Computer Science Institute, UNICAMP, Brazil, 2007.

[7] Arnaldo Jovanini Montagner and Jorge Stolfi. Gems: A general data structure for d-dimensional triangulations. *Technical Report IC-06-16, Computer Science Institute, UNICAMP*, 2006.

[8] Michael Spivak. *Calculus on manifolds. A modern approach to classical theorems of advanced calculus.* W. A. Benjamin, Inc., New York-Amsterdam, 1965.

[9] Richard Steinberg. The state of the three color problem. In *Quo Vadis, Graph Theory? A Source Book for Challenges and Directions*, volume 55 of *Annals of Discrete Mathematics*, pages 211 – 248. Elsevier, 1993.