Delaunay mesh generation with triangle area constrains *

Narcís Coll, Marité Guerrieri and J. Antoni Sellarès †

Abstract

We propose a framework that combines Delaunay refinement and improvement techniques for generating a refined Delaunay quality mesh of a Planar Straight Line Graph domain. Our algorithms achieve quality mesh by moving and inserting Steiner points from or into the mesh, combining the Delaunay criterion with different area-based criteria.

1 Introduction

The discretization of a continuous planar domain is an essential step in numerical simulation of physical and engineering problems where numerical methods, such as the finite element method, are used. A triangular mesh is a discretization of the domain into triangles that intersect only at shared edges and vertices. Each triangle of the mesh must be well shaped in order to achieve the convergence of those numerical methods.

There are several shape criteria, being the most common the Delaunay criterion, that maximizes the minimum of the angles of the triangles of the triangulation, ensuring that the angles of each triangle are neither too small nor too large. Maximizing the minimum area of mesh triangles without taking into account the angles is not ideal for mesh generation because, in most cases, the resulting mesh can have many long and skinny triangles. Nevertheless, Surazhsky at ed. in [6] showed that a triangulation having triangles with areas close to equal has one important property: the distribution of the mesh vertices is very uniform. For this reason area-based criteria are used for mesh control and improvement.

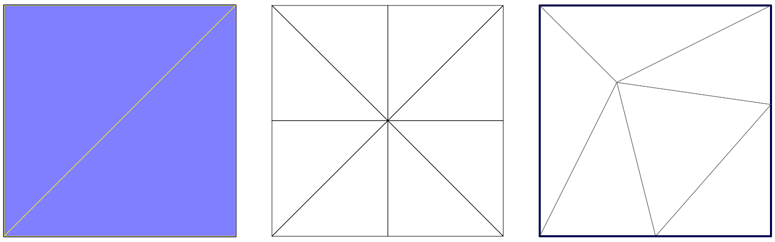
There exist many works on the generation of a quality triangular mesh for a Planar Straight Line Graph (PSLG) domain. Delaunay refinement mesh generation algorithms have taken place in this context [16, 3, 17, 14, 20, 10]. In keeping quality of a mesh two objectives are pursued. First, force segments of the PSLG into the mesh. Second, get skinny triangles, triangles without the required quality, out of the mesh. Both goals are achieved by the addition of Steiner points, points that do not belong to the original mesh. The former objective is carried out by the addition of midpoints on constrained segments to insert. Meanwhile, dealing with the later goal, several works studied the problem of where to insert these additional vertices into an existing mesh. Among them, the algorithms in [16, 3] use circumcenters of poor quality triangles. Recently has been proposed a new algorithm [20] that inserts specially chosen points, the so-called off-centers. The algorithm generates quality-guaranteed size-optimal meshes and an experimental study indicates that inserts considerably fewer Steiner vertices than algorithms which insert circumcenters.

The improvement of a mesh is based on local optimizations that involve the movement of Steiner vertices and the rearrangement of the triangulation. In the smoothing technique Steiner vertices are moved to improve the quality of their adjacent elements. The movement of vertices lying on PSLG segments can be forbidden or restricted to the segment itself. A smoothing algorithm is applied several times through the entire set of vertices of the mesh. The most simple smoothing technique is Laplacian smoothing, in which a vertex is moved to the centroid of the vertices to which it is connected [11], but it has the drawback that an invalid mesh can be produced. In the last years several smoothing techniques, based on optimization, that avoid the creation of invalid elements and guarantee

*Partially supported by the Spanish Ministerio de Educación y Ciencia under grant TIN2004-08065-C02-02.
†Departament d'Informàtica i Matemàtica Aplicada, Universitat de Girona, {coll,mariteg,sellares}@ima.udg.es

an optimal location had been proposed [1, 15, 18]. Freitag at ed. in [7, 8] proposed a parallel algorithm based on optimization. In [9] Freitag presents several techniques that combine the low cost Laplacian smoothing with the optimization-based approach used only for the poorest quality elements. Works [6, 5] add Steiner points to the mesh until all the triangles achieve some area requirements. A new method which combines optimization and smoothing techniques in the generation and modification of Delaunay refined meshes is presented in [2].

In the following we give a simple example to introduce the interesting problem of obtaining an area bound adding as few Steiner points as possible. Figure 1(a) shows an initial mesh with triangles whose area is greater than a certain bound. In the middle, 1(b), the result of adding any number of Steiner points to fulfill the objective of not having any triangle in the mesh with area greater than the established bound, and in the right, 1(c), a better result with a lower number of Steiner points.



(a) (b) (c)

Figure 1: (a) A Delaunay triangulation of the PSLG composed by a square boundary of 3x3 cm with triangles with area greater than 1.0 painted in the figure. (b) A Delaunay triangulation with all its triangles with area less or equal than 2.0. (c) Another Delaunay triangulation which fulfill the two requirements of triangles with area less or equal than the established bound and fewer Steiner points than that obtained in the previous figure. algorithm.

In this paper we present a framework that combines the Delaunay criterion with different area-based criteria, obtaining a trade-off between improving the area-based quality and decreasing the minimum-angle quality.

2 Preliminaries

A *Planar Straight Line Graph* (PSLG) is a set of points and segments satisfying two constraints: all endpoint segments are points in the PSLG, segments may intersect each other only at their endpoints.

A triangulation $\mathcal{T}$ is a *conforming triangulation* of a PSLG, Ω, if each *point* in Ω corresponds to a *vertex* in $\mathcal{T}$, and a *segment* of Ω is represented by a linear contiguous sequence of *edges* of $\mathcal{T}$. New Steiner vertices, not points of Ω, may appear, and each segment of Ω may have been subdivided into shorter edges by these additional vertices. Flipping these edges is forbidden, then they are marked as *locked*. In a *conforming Delaunay triangulation* of a PSLG, the Steiner vertices are added so that the Delaunay property is maintained.

The *star* of a vertex q, $\mathcal{S}_q$, of a triangulation $\mathcal{T}$ consists of all the triangles of $\mathcal{T}$ that contain q. The *link* of q, $\mathcal{L}_q$, is the polygon determined by the set of edges of the triangles in $\mathcal{S}_q$ that are not incident to q. Since the average degree of a node in a planar graph is less than six [3], the average number of triangles of $\mathcal{S}_q$ or the average number of edges of $\mathcal{L}_q$, is at most six.

Given an edge $e \in \mathcal{L}_q$, whose endpoints are e_1 and e_2, we use the following notation (see Figure 2):

- $H_{q,e}$ is the open half-plane determined by e and containing the vertex q.
- $t_{q,e}$ is the triangle with vertices e_1, e_2 and q.
- $t'_{q,e}$ is the adjacent triangle to $t_{q,e}$ by e.
- $c_{q,e}$ is the circumcircle of $t'_{q,e}$.
- $a_{q,e}$ is the arc $c_{q,e} \cap H_{q,e}$. We will say that $c_{q,e}$ is the supporting circle of $a_{q,e}$.

- $A_{q,e}$ is the area of a triangle $t_{q,e}$.

A triangle $t_{q,e}$ having area $A_{q,e}$ greater than α, for certain fixed α, is called a *bad* triangle.

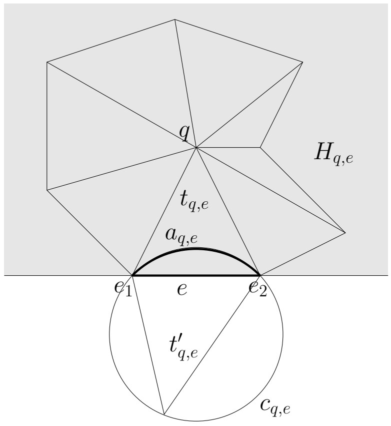


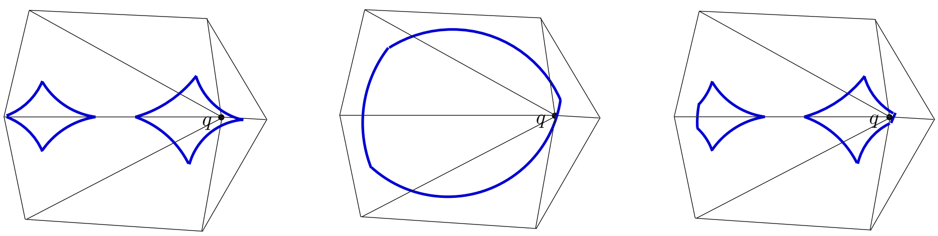
Figure 2: Notation used in the definitions.

The *kernel* of $\mathcal{L}_q$, denoted by $ker(\mathcal{L}_q)$, is the set of all points $p \in \mathcal{L}_q$, such that for every vertex v of $\mathcal{L}_q$, the segment vp is within $\mathcal{L}_q$. It can be proved that $ker(\mathcal{L}_q) = \bigcap_{e \in \mathcal{L}_q} H_{q,e}$.

The *Delaunay zone of an edge* $e \in \mathcal{L}_q$, denoted $\mathcal{D}_{q,e}$, is the set of points of $H_{q,e}$ external to $c_{q,e}$ (see Figure 2). The *external Delaunay zone of a vertex* q is the set $\mathcal{ED}_q = \bigcap_{e \in \mathcal{L}_q} \mathcal{D}_{q,e}$. The external Delaunay zone of a vertex is an open non-convex set included in $ker(\mathcal{L}_q)$ and, as exhibited in Figure 3(a), may be constituted by several non-connected components.

Let $C_{\mathcal{L}_q}$ be the convex vertices of $\mathcal{L}_q$. The *Delaunay zone of a vertex* $u \in C_{\mathcal{L}_q}$, denoted $\mathcal{D}_{q,u}$, is the interior of the circle determined by u and its adjacent vertices in $\mathcal{L}_q$. The *internal Delaunay zone of a vertex* q is the convex set $\mathcal{ID}_q = \bigcap_{u \in C_{\mathcal{L}_q}} \mathcal{D}_{q,u}$ (see Figure 3(b)).

The *Delaunay zone of a vertex* q is the set $\mathcal{D}_q = \mathcal{ED}_q \cap \mathcal{ID}_q$ (see Figure 3(c)). The boundary of $\mathcal{D}_q$ will be denoted by $\overline{\mathcal{D}_q}$. If we replace q by a point $p \in \mathcal{D}_q$ then $\mathcal{L}_p = \mathcal{L}_q$ and the set of triangles $\{t_{q,e}\}$ is substituted by the set $\{t_{p,e}\}$.



(a) (b) (c)

Figure 3: (a) External Delaunay zone. (b) Internal Delaunay zone. (c) Delaunay zone.

2.1 Incremental Delaunay algorithm

There exists three types of algorithms for constructing Delaunay triangulations, namely, divide-and-conquer, sweepline and incremental. We concentrate our attention in the latter ones.

Incremental algorithms add vertices one by one and update the triangulation after each vertex is added maintaining the Delaunay property. The original algorithm, developed by Lawson [4], is based upon edge flips. There are incremental algorithms due to Bowyer [1] and Watson [7] that do not use edge flips. In Lawson's algorithm, when a vertex is inserted, the triangle that contains it is found, and three new edges are inserted to attach the new vertex to the vertices of the containing triangle. If the new vertex falls upon an edge of the triangulation, that edge is deleted, and four new edges are inserted to attach the new vertex to the vertices of the containing quadrilateral. Next, a recursive procedure tests whether the new vertex lies within the circumcircles of any neighboring triangles, each affirmative test triggering an edge flip that removes a locally non-Delaunay edge. Each edge flip reveals two additional edges that must be tested.

2.2 Lower and upper envelopes of a set of planes

Let Π a set of non-vertical planes. The *lower envelope*, $\mathcal{LE}_\Pi$, of the set of planes Π is the polyhedral surface that bounds the "lowermost" unbounded cell of the arrangement defined by the planes of Π. If we regard each plane $\pi \in \Pi$ as the graph of a linear function $\pi(p)$, where p is a point of the XY-plane, the lower envelope $\mathcal{LE}_\Pi$ is the graph of the piecewise concave function:

$$\mathcal{LE}_\Pi(p) = \min_{\pi\in\Pi}\{\pi(p)\}.$$

The *upper envelope*, $\mathcal{UE}_\Pi$, of the set of planes Π is the polyhedral surface that bounds the "uppermost" unbounded cell of the arrangement defined by the planes of Π. The upper envelope $\mathcal{UE}_\Pi$ is the graph of the piecewise convex function:

$$\mathcal{UE}_\Pi(p) = \max_{\pi\in\Pi}\{\pi(p)\}.$$

The set of vertices of $\mathcal{LE}_\Pi$ and $\mathcal{UE}_\Pi$ are denoted $V_{\mathcal{LE}_\Pi}$ and $V_{\mathcal{UE}_\Pi}$, and their projection onto the XY-plane $\overline{V}_{\mathcal{LE}_\Pi}$ and $\overline{V}_{\mathcal{UE}_\Pi}$, respectively.

We denote $\mathcal{UE}_\Pi - \mathcal{LE}_\Pi$ the graph of the piecewise convex function $\mathcal{UE}_\Pi(p) - \mathcal{LE}_\Pi(p)$. The set of projections of the vertices of $\mathcal{UE}_\Pi - \mathcal{LE}_\Pi$ onto the XY-plane is $\overline{V}_{\mathcal{LE}_\Pi} \cup \overline{V}_{\mathcal{UE}_\Pi}$.

The polyhedral surfaces $\mathcal{LE}_\Pi$, $\mathcal{UE}_\Pi$ and $\mathcal{UE}_\Pi - \mathcal{LE}_\Pi$ have $O(n)$ complexity and can be obtained in $O(n \log n)$ time, where n is the number of planes of Π.

3 Finding a point optimizing a star

Let $\mathcal{T}$ be a Delaunay refined triangulation of a PSLG. Consider a vertex q of $\mathcal{T}$. We are interested in replacing the point q by a point $\widetilde{p} \in D_q$, to assure $\mathcal{L}_p = \mathcal{L}_q$, such that optimizes diverse criteria related to the area of the triangles of the star $\mathcal{S}_q$.

3.1 Problem transformation

From now on we identify $\mathbb{R}^2$ with the xy-plane of $\mathbb{R}^3$ so that the point $(x, y) \in \mathbb{R}^2$ becomes the point $(x, y, 0) \in \mathbb{R}^3$. We also assume that the triangulation $\mathcal{T}$ is located on the xy-plane and that the edges of any star $\mathcal{S}_q$ are ordered counterclockwise.

If the coordinates of the endpoints of and edge e of $\mathcal{L}_q$ are $e_1 = (e_{11}, e_{12}, 0)$ and $e_2 = (e_{21}, e_{22}, 0)$, then for any point $p = (p_1, p_2, 0) \in \mathcal{S}_q$ we have:

$$A_{p,e} = \frac{1}{2}\begin{vmatrix} e_{21} - e_{11} & p_1 - e_{11} \\ e_{22} - e_{12} & p_2 - e_{12} \end{vmatrix}. \tag{1}$$

Consequently, the area $A_{p,e}$ is a linear function of $p \in \mathcal{S}_q$, denoted $\pi_e(p)$, whose graph is (a bounded part) of the plane π_e through the segment e with normal vector $(-e_{22} + e_{12}, e_{21} - e_{11}, 2)$.

Let Π_q be the the set of all planes π_e, $e \in \mathcal{L}_q$. Then we have:

$$\min_{e\in\mathcal{L}_q}\{A_{p,e}\} = \min_{\pi_e\in\Pi_q}\{\pi_e(p)\} = \mathcal{LE}_{\Pi_q}(p) \qquad \text{and} \qquad \max_{e\in\mathcal{L}_q}\{A_{p,e}\} = \max_{\pi_e\in\Pi_q}\{\pi_e(p)\} = \mathcal{UE}_{\Pi_q}(p). \quad (2)$$

3.2 Optimization problems

We want to find $\widetilde{p} \in \mathcal{D}_q$ that satisfies one of the following criteria:

- $\widetilde{p}$ maximizes the minimum area among the triangles of $\mathcal{S}_q$:

$$\max_{p\in\mathcal{D}_q}(\min_{e\in\mathcal{L}_q}\{A_{p,e}\}) = \max_{p\in\mathcal{D}_q}\mathcal{LE}_{\Pi_q}(p). \quad (3)$$

- $\widetilde{p}$ minimizes the difference between the maximum and minimum area among the triangles of $\mathcal{S}_q$:

$$\min_{p\in\mathcal{D}_q}(\max_{e\in\mathcal{L}_q}A_{p,e} - \min_{e\in\mathcal{L}_q}A_{p,e}) = \min_{p\in\mathcal{D}_q}(\mathcal{UE}_{\Pi_q}(p) - \mathcal{LE}_{\Pi_q}(p)). \quad (4)$$

3.3 Optimization algorithms

The method to find these points is based on the following properties:

- Any local minimum (maximum) of a convex (concave) function defined on a compact set is also a global minimum (maximum).
- If a continuous function defined on a compact set does not have any local extremum, their global extrema lies on its boundary.
- A linear function $z = Ax + By + Cz + D$ defined on a circle of center (c_1, c_2) and radius r reaches its maximum on the point $\left(c_1 + \frac{Ar}{\sqrt{A^2+B^2}},\ c_2 + \frac{Br}{\sqrt{A^2+B^2}}\right)$ and its minimum on the point $\left(c_1 - \frac{Ar}{\sqrt{A^2+B^2}},\ c_2 - \frac{Br}{\sqrt{A^2+B^2}}\right)$.

Our method first finds the local extremum p^* that optimizes the area of the triangles of $\mathcal{S}_q$. If $p^* \in \mathcal{D}_q$ then p^* is the optimal solution $\widetilde{p}$. Otherwise we find the point $\widetilde{p}$ that optimizes the areas of the star restricted to $\overline{\mathcal{D}_q}$ and we take a point inside $\mathcal{D}_q$ close to $\overline{p}$ as the target point $\widetilde{p}$. This process needs to compute the arcs of $\overline{\mathcal{D}_q}$ and the envelopes $\mathcal{LE}_{\Pi_q}$ and $\mathcal{UE}_{\Pi_q}$. Since the computation depends on the number of edges of $\mathcal{L}_q$ whose average number is six, the time needed for the overall method is constant.

3.3.1 First step

The first step that finds a point p^* on $\mathcal{S}_q$ satisfying one of the two area criteria established is explained in this subsection.

Finding a point maximizing the minimum area (see Figure 4)

1. Determine Π_q.
2. Find $\mathcal{LE}_{\Pi_q}$.
3. Determine V the vertices of $\mathcal{LE}_{\Pi_q}$.
4. Let p^* be the vertex in V with the maximum height.

Finding a point minimizing the difference between areas (see Figure 5)

1. Determine Π_q.

2. Find $\mathcal{LE}_{\Pi_q}$ and $\mathcal{UE}_{\Pi_q}$.
3. Determine V_{LE}, V_{UE}, $\overline{V}_{LE_{\Pi_q}}$ and $\overline{V}_{UE_{\Pi_q}}$.
4. Determine $V = \{(p_1, p_2, \mathcal{LE}_{\Pi_q}(p) - \mathcal{UE}_{\Pi_q}(p)) \mid (p_1, p_2) \in \overline{V}_{LE_{\Pi_q}} \cup \overline{V}_{UE_{\Pi_q}}\}$.
5. Let p^* be the vertex of V with the minimum height.

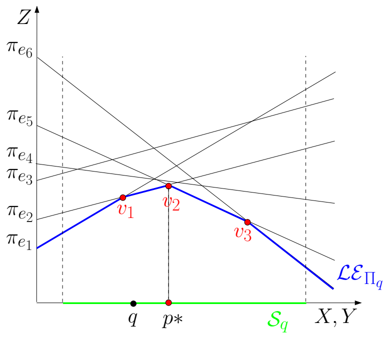


Figure 4: First step in finding a point maximizing the minimum area.

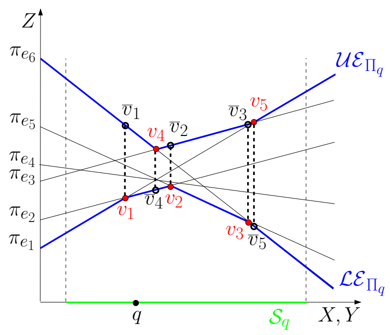


Figure 5: First step in finding a point minimizing the difference between areas.

3.3.2 Second step

The second step finds a point $\widetilde{p}$ on $\overline{\mathcal{D}_q}$ (see Figure 6). Let $\overline{F}$ be the set of the projected faces of the polyhedral surface considered ($\mathcal{LE}$ or $\mathcal{UE} - \mathcal{LE}$). The point $\widetilde{p}$ is found by applying the following steps:

1. Compute the arcs a_i of $\overline{\mathcal{D}_q}$.
2. Split each arc a_i in subarcs a_{ij} according to the faces of $\overline{F}$ intersected by a_i (see Figure 6(a)).
3. For each subarc a_{ij} compute the global extremum point p_{ij} of the linear function associated to the face intersected by a_{ij} (see Figure 6(b)).
4. Let $\overline{p}$ be the point p_{ij} that optimizes the criterion considered and $\widetilde{p}$ be a point inside $\mathcal{D}_q$ close to $\overline{p}$ (see Figure 6(c)).

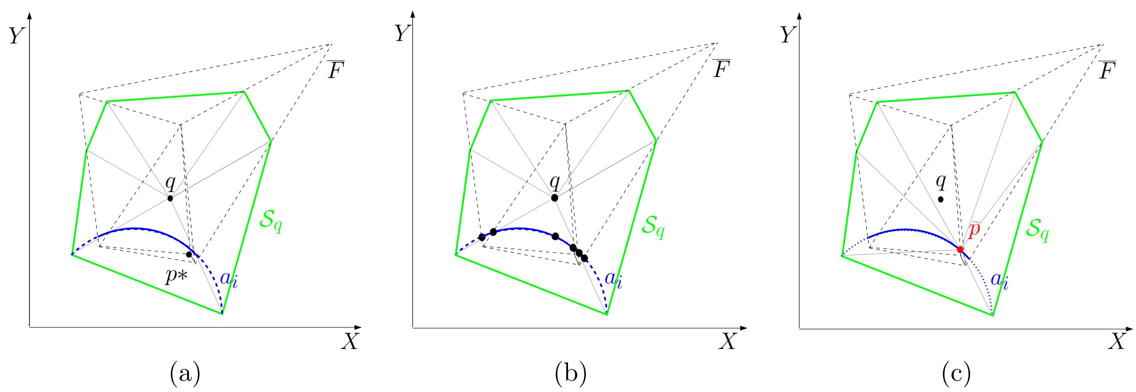


Figure 6: Second step in finding a point into the Delaunay zone.

4 Movement of Steiner points

Movement of Steiner points is the basic operation used by our improvement process. Steiner points to be treated by the process can belong to two main groups. The first group is formed by the Steiner points located on any segment of the PSLG, and the second group is formed by the remaining Steiner points. We name *restricted* vertices the points of the first group, since their movement will be restricted to the corresponding segment, and *free* vertices the points of the second group.

4.1 Moving free vertices

The key concept regarding the movement of a free vertex q is to substitute this vertex by the best point in $\mathcal{S}_q$ such that it optimizes the area criterion of the triangles of the star, being α the area quality requirement of the mesh. In order to do that, we apply the optimizing algorithm explained in the previous section.

4.2 Moving restricted vertices

The movement of restricted vertices is constrained over their correspondent subsegments. This kind of vertices can be present on a boundary subsegment or on a non-boundary subsegment of a PSLG. Since the optimizing algorithm can easily be adapted in order to guarantee that the vertex $\widetilde{p}$ lies on the subsegment, in both cases we apply the iterative process explained in Section 3.3.

5 Generating a Delaunay refined mesh

The process to generate a refined Delaunay quality mesh of a PSLG considering the area of its triangles as the quality measure consists of the following steps. First, a conforming Delaunay triangulation of the PSLG is generated, then the list of bad triangles (area bigger than α) to be removed is obtained, and finally our improvement method is applied to eliminate those bad triangles. Observe that initially the list of points to be inserted is empty.

5.1 Improvement process

The improvement process receives as input the list of bad triangles to be removed. The output of the process is a mesh with the desired quality. The process maintains this list and finishes when it is empty. To remove a bad triangle, first its vertices are checked for movement, if it is not possible the midpoint of its longest edge is inserted and moved to an optimal position.

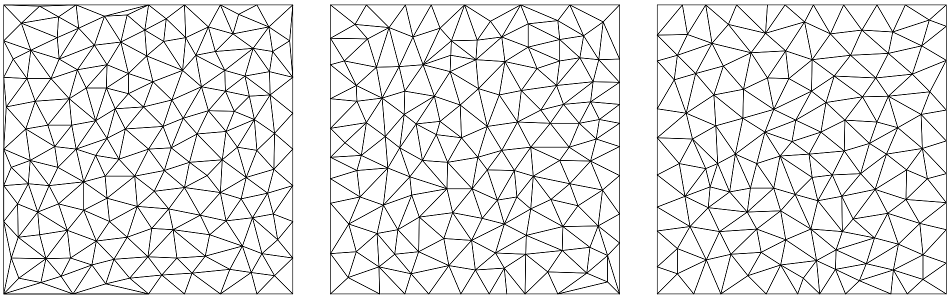
6 Experimental results

We have implemented our algorithms in C++ language and using OpenGL libraries to build an interactive interface. Our application takes a triangulated PSLG as input. This initial mesh is refined until the desired area quality is achieved. In the optimization method we use the freely available software *Qhull* [9] to calculate the needed envelopes.

We have run several simulations in order to test our implementation and we have compared these simulations with meshes generated using *Triangle*, a freely available software produced by Jonathan R. Shewchuck [8], which offers the possibility of generating a mesh with a required area quality. We work results on two PSLGs. The first PSLG is the 3x3 cm square presented in the introduction, and the second one is composed of a square boundary and a polygonal hole described by 20 points. The results obtained are presented in Figures 7 and 8. It can be observed in the results obtained that the number of triangles generated by *Triangle* are higher than applying our algorithm. Also notice that the second criterion produce less triangles than the first criterion.

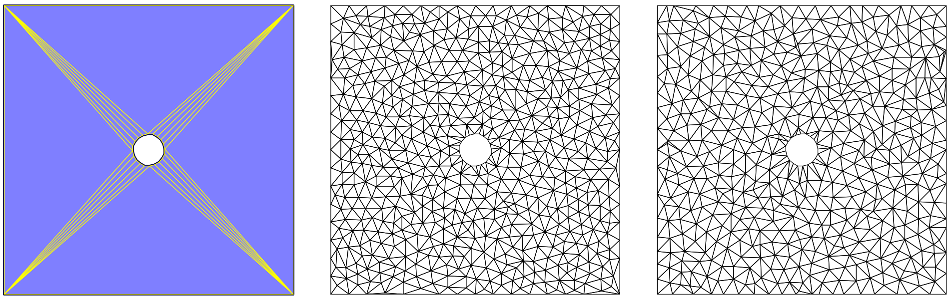
7 Future work

Future work include the study of the deletion of Steiner vertices in order to reduce the number of triangles while maintaining the area quality. We also are interested in to apply the algorithm developed when the a Delaunay refined mesh is modified. Modify a mesh means to insert new PSLG elements into the mesh or delete PSLG elements from the mesh.



(a) 287 triangles. (b) 256 triangles. (c) 235 triangles

Figure 7: Results obtained in the generation of a mesh for an area of 0.05. In (a) the result from *Triangle*, in (b) and (c) applying our algorithm maximizing the minim area and minimizing the difference between areas, respectively.



(a) Initial mesh. (b) 1147 triangles. (c) 943 triangles.

Figure 8: Results obtained in the generation of a mesh for an area of 0.05. In (a) the initial mesh with bad triangles painted, in (b) the result from *Triangle* and in (c) applying our algorithm minimizing the difference between areas.

Referencias

[1] A. Bowyer. Computing Dirichlet Tessellations. *Computer Journal*, 24:2:162–166, 1981.

[2] N. Coll and M. Guerrieri and J.-A. Sellarès. Mesh modification under local domain changes. *In Proceedings of 15th International Meshing Roundtable*, 39–56, 2006.

[3] K. Clarkson and K. Mehlhorn and R. Seidel. Four results on randomized incremental constructions. *Computational Geometry: Theory and Applications*, 3:185–212, 1993.

[4] C.-L. Lawson. Software for $\mathcal{C}^1$ Surface Interpolation. *Mathematical Software III(John R.Rice, editor)*, 161–194, 1977.

[5] J.-R. Shewchuk. Delaunay Refinement Mesh Generation. *School of Computer Science - Carnegie Mellon University - PhD thesis*, 1997.

[6] V. Surazhsky and C. Gostman. High quality compatible triangulation. *In Proceedings 11th International Meshing Roundtable*, 183–192, 2002.

[7] D.-F. Watson. Computing the n-dimensional Delaunay Tessellation with Application to Voronoi Polytopes. *Computer Journal*, 24:2:167–172, 1981.

[8] http://www.cs.cmu.edu/~quake/triangle.html.

[9] http://www.qhull.org/