A survey of point insertion techniques in bidimensional Delaunay Triangulations

S. L. G. de Oliveira

UFLA - Universidade Federal de Lavras
DCC - Departamento de Ciência da Computação
P.O. Box 3037 - Campus da UFLA 37200-000 - Lavras (MG) - Brazil
sanderson@dcc.ufla.br

Abstract. Triangulations are geometric discretizations essential in many scientific applications, such as engineering simulations, visualizations, and geographic information systems. The preferred shape of a triangle depends on the applications. Theoretical and experimental analysis of numerical methods that are used in conjunction with triangulations suggest that triangles with no large angles and/or small angles serve well in most applications. This paper is a brief review of a point insertion in 2D Delaunay Triangulations. Important works on the insertion of vertices in Delaunay Triangulations are described as a start point for one who needs to build a quality mesh using adaptive triangular-mesh refinement.

Keywords: Delaunay Triangulation, mesh generation, adaptive triangular mesh refinement, computational geometric modeling.

(Received April 10th, 2011 / Accepted June 18th, 2011)

1 Introduction

Triangulations are geometric discretizations essential in many scientific applications, such as engineering simulations, medical imaging, visualizations, and geographic information systems [22]. Erten and Üngör [22] explain that the preferred shape of a triangle depends on the applications. However, theoretical and experimental analysis of numerical methods that are used in conjunction with triangulations suggest that triangles with no large angles and/or small angles serve well in most applications (see [1]). According to Erten and Üngör [22], "in general, the better the shape of the triangles, the smaller the interpolation and approximation errors are in their use".

Delaunay triangulations are optimum in maximizing the smallest angle [17]. An approach in order to provide quality triangular meshes is to use algorithms based on a automatic point insertion strategy on the Delaunay Triangulation. A planar Delaunay Triangulation [15] for a point set P is a triangulation DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P). The Delaunay Triangulation builds the optimal triangular mesh. This means that it builds triangles more similar to the equilateral ones for a given fixed point set.

The Delaunay Triangulation and its duals Voronoi Diagram [52] and medial axes have been applied in many different fields, such as the ones earlier cited, including numerial methods and computer graphics. The reader is referred to Guibas and Stolfi [26] and Barth [4] for properties and algorithms in order to build 2D Delaunay Triangulations. Shewchuk [45] presented aspects of the Delaunay mesh generation. Edelsbrunner [18] provided a theoretical review on Delaunay Triangulation. De Floriani and collaborators [12] reviewed the basic triangulation properties, Delaunay Triangulations, constrained and conforming triangulations. They also presented a survey of algorithms for building these kind of triangulations, mainly in the context of digital terrain modeling in geographic information systems.

In order to build a Delaunay Triangulation, the reader is referred to the mesh generation software Triangle [48]. Triangle's high-quality mesh generation is based on Chew-Ruppert Delaunay refinement algorithm [41]. Both were surveyed by Shewchuk in [46]. In addition, Shewchuk described Ruppert's Delaunay refinement algorithm in [47]. These algorithms evolved from the works of Chew [7] and Bern *et al.* [5]. The Chew-Ruppert Delaunay refinement method is modified in Triangle to handle domains with small angles well, following a idea in the paper of Miller *et al.* [32]. It also incorporates a modification by Üngör [51] that reduces the number of triangles generated. Triangle's implementation of the divide-and-conquer and incremental Delaunay triangulation algorithms follows closely the presentation of Guibas and Stolfi [26]. Triangle uses a triangle-based data structure instead of Guibas and Stolfi's quad-edge data structure. The $O(n \log n)$ divide-and-conquer algorithm promoted by Guibas and Stolfi was originally developed by Lee and Schachter [31]. Dwyer [16] showed that the algorithm is improved by using alternating vertical and horizontal cuts to divide the vertex set. Triangle uses an expected $O(n^{1/3})$ time point location scheme proposed by Mücke [33]. Triangle's $O(n \log n)$ sweepline algorithm for Delaunay triangulation is due to Fortune [23], and relies upon Sleator and Tarjan's splay trees [50]. The earlier description is based on the Triangle's website [48].

Given a Delaunay Triangulation, one is allowed to insert points (called the Steiner points) in order to compute good quality triangulations. This, however, may increase the number of points and triangles in a triangulation, which is a key factor in the running time of an application algorithm. The reader is referred to [22] for details and a survey on the context of providing a good triangulation.

After this brief introduction, Section 2 provides a further review of the schemes for point insertion in a Delaunay Triangulation in the context of providing a adaptive refined mesh. Section 3 describes the Voronoi Diagram. Section 4 surveys the Rivara's schemes and others. Some future directions are given in Section 5.

2 Point insertion in a Delaunay Triangulation

A point insertion in a Delaunay Triangulation is not a trivial task. For example, if one simply inserts a point into the triangle barycenter (Figure 1a), this process fastly degenerates the triangulation quality, specially along boundaries. This occurs even when carrying out global refinement. In [38], the authors affirm that a pure Delaunay algorithm does not provide a natural point insertion scheme that guarantees the construction of good-quality nonuniform triangulations when the algorithm is iteratively used in the adaptive mesh refinement. They described experiments with the simple centroid insertion (see Figure 1a) concept.

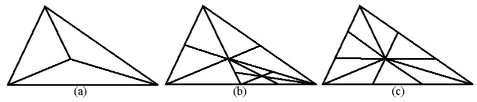

Figure 1: Triangle partition processes: (a) *ternary subdivision* - refinement by simple centroid insertion; (b) refinement by centroid insertion and adding midedges - a second refinement is performed in the bottom right triangle; (c) trisection of the edges, joinning the centroid to those points and also to the vertices.

The literature is rich in approaches to introduce points into the triangulation. These schemes provide high-quality Delaunay Triangulations and some of them are described in the following.

Fowler and Little [24] proposed the *vertex insertion* in conjunction with the Delaunay Triangulation. A Delaunay criterion localizes the position of a potential point to be inserted. This could affect the fit to the circumscribed circle about the triangle. The authors argued that it is sufficient to perform series of domain-limited searches in each triangle of the model; rather than carrying out global searches for the global "worst-fit" points. In this approach, adding a point destroys the original triangle and introduces new triangles. The inserted point is a vertex of the new triangles. In Figure 2, a point is introduced and the region is triangulated. The reverse operation, known in computer-graphic context as *decimation*, is performed in order to unrefine the region. In a variation, a point is inserted, the set of triangles on its neighborhood are deleted and the region is retriangulated (Figure 3). The inverse operator, the *vertex removal*, deletes a point together with its incident triangles and constructs new triangles in the region.

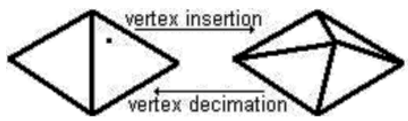

Figure 2: Vertex insertion and vertex decimation.

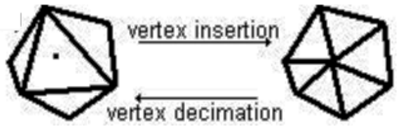

Figure 3: Vertex insertion and vertex decimation.

Clarkson and Shor [9] showed that if the order of vertex insertion is randomized, each vertex can be inserted in $O(n)$ time, not counting point location (see details in Shewchuk [48]. Chew ([7] and [8]) proposed a Delaunay improvement algorithm that triangulates a given polygon into a uniform mesh with all angles between 30 and 120. It guarantees that the output mesh is size-optimal within a constant factor amongst all uniform meshes.

The Hierarchical Delaunay Triangulations (HDT) was proposed by De Floriani and Puppo in [13] and [14]. It is based on a hierarchy of triangle-based surface approximations, where each node, except the root, is a triangulated irregular network refining a triangle face belonging to its parent in the hierarchy (see Figure 4). This method is similar to the proposed by Scarlatos Pavlidis in [42]; however, the triangle subdivision is more general. The subdivision inside every macro-triangle is locally a Delaunay Triangulation; whereas a global expanded subdivision of the whole domain gerally is not. The triangle partition is performed by an iterative application of a selector process that, at each step, updates the current Delaunay Triangulation by introducing the point having the maximum error. Moreover, in order to subdivide a triangle for a given hierarchical level, they used a curve approximation algorithm [3] in order to insert points along the edges. Afterwards, points are added in the inner triangle until an error threshold is met throughout the triangle. So, the inner triangle is retriangulated using Delaunay Triangulation. The constructing algorithm basis for a HDT must be an on-line approach that incrementally builds a Delaunay Triangulation through iterative point insertion [12]. According to Heckbert and Garland [28], the HDT seems to present nearly identical flexibility and speed compared to the one proposed in [42]. However, for a given error threshold, the HDT likely yields slightly better simplification.

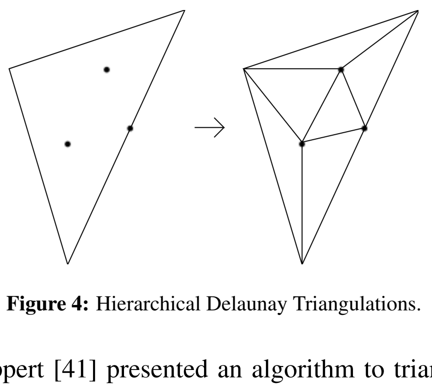

Figure 4: Hierarchical Delaunay Triangulations.

Ruppert [41] presented an algorithm to triangulate planar straightline graphs. It guarantees that every triangle in the output mesh has smallest angle greater than 278. It produces a size-optimal nonuniform mesh. It is also size-optimal to within a constant factor. The idea behing these algorithms is either: to refine a small angled triangle by the Delaunay insertion of its circumcenter; or a modification of the boundary if the circumcircle is external to the meshing region (see [37] for details). Baker [2] published a comparison of edge and circumcenter based refinements. Properties of mesh improvement for iterative Delaunay refinement based on inserting a point in the circumcenter of triangles to be refined was also established by Shewchuk in [44]. A combination of edge refinement and Delaunay point insertion was described by Borouchaki and George in [6] and [25].

Shewchuk [46] presentd a framework for analyzing Delaunay refinement algorithms that unifies the mesh generation algorithms of Chew and Ruppert. The Shewchuk's framework improves the Chew's and Ruppert's algorithms in several ways, and also helps to solve the difficult problem of meshing nonmanifold domains with small angles.

Üngör [51] presented an algorithm based on the *off-center* insertion. In the former case, the off-center of a triangle with the shortest edge $\overline{pq}$ is a point o on the bisector of $\overline{pq}$ furthest from p (or q) such that the angle among the three points is a user-specified constraint angle. The idea of using *off-centers* led Har-Peled and Ungor [27] to the design of the first time-optimal Delaunay refinement algorithm.

Erten and Üngör [21] proposed algorithms that improve the off-center performance with respect to the mesh size and a minimum angle tolerance. This is performed by using point selections depending on some triangle cases. Erten and Üngör [20] published a Delaunay refinement algorithm that generally terminates for constraint angles up to 42°.

Erten and Üngör [22] proposed two algorithms to improve the performance of Delaunay refinement. The first one uses the Voronoi Diagram and unifies previously suggested Steiner point insertion schemes (circuncenters [7], [40], [46], sink [19], off-center [51]) together with a proper strategy. The second algorithm integrates a local smoothing strategy into the refinement process. For a given input domain and a constraint angle α, the Delaunay refinement algorithms aim to compute triangulations with angles at least α.

Recently, Plaza and collaborators [34] proposed the 7-triangle Delaunay partition (Figure 5). This refinement scheme also propagates the refinement and inserts non-similar triangles.

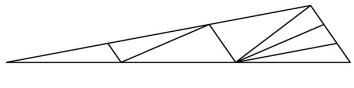

Figure 5: 7-triangle Delaunay partition

3 Voronoi Diagram

The Voronoi Diagram was proposed in [52]. Shamos [43] was the first to argue that the Voronoi Diagram can be used as a tool to provide efficient algorithms for a wide variety of geometric problems.

Barth [4] defined the Delaunay Triangulation of a point set as the dual of the Voronoi Diagram of the set. The 2D-Delaunay Triangulation is formed by connecting two points if and only if their Voronoi regions have a common border segment. If no four or more points are cocircular, then the vertices of the Voronoi Diagram are the triangle circumcenters. Moreover, Voronoi vertices represent locations that are equidistant to three or more points.

Consider the Delaunay Triangulation of a set *V* of planar points. The Voronoi Diagram describes the proximity relationship among the points of *V*. The Voronoi Diagram of a set *V* of *n* points is a planar subdivision into *n* convex polygonal regions. Each region is associated with a point of *V*. Each Voronoi region of each point of *V* is the set of planar points which lie closer to the point than to any other point in *V*. Two points of *V* are neighbors when the corresponding Voronoi regions are adjacent [12].

An interface orthogonal to the segment between two centroids facilitates finite-volume approximations. Moreover, it improves the solution accuracy and reduces the computational effort to approximate a solution of a partial differential equation. Furthermore, in this approach, the finite volumes are not the triangles themselves, but the Voronoi Diagram (see Figure 6), i.e. parts of each triangle.

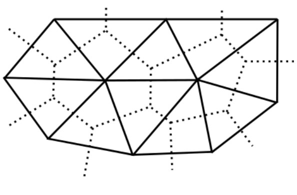

Figure 6: A single Delaunay Triangulation and its dual the Voronoi Diagram.

4 Longest-edge based triangle partition within Delaunay Triangulation

Rivara [35] presented the backward longest-edge refinement (BLER) algorithm based on an interesting concept in order to conform the mesh in the finite element context: the longest-edge propagation path (LEPP). Briefly, the LEPP keeps a path of *n* triangles that have also to be refined for each triangle of the mesh. For example, consider that the triangle t_0 is marked to be refined. The LEPP indicates that the triangles $t_1, t_2, \cdots, t_n$ also must be refined in order to mantain a conforming good-quality mesh. It propagates the list until the longest-edge shared by triangles t_{n-1} and t_n. This edge is larger than the one of its previous neighbor or t_n is in the boundary. Figure 7 shows an example of the LEPP-midedge propagation with 3T-LE partition approach and t_n is bisected, where *n*=4 in this example. The BLER is a partition procedure that extended both the pure longest-edge refinement algorithms for general nonDelaunay Triangulation (see [39] and the references therein) and the longest-edge refinement algorithm for Delaunay Triangulations proposed by Rivara and Inostroza [38]. Specifically, the algorithm presented in [38] guarantees that meshes of analogous quality to the input reference-mesh are built.

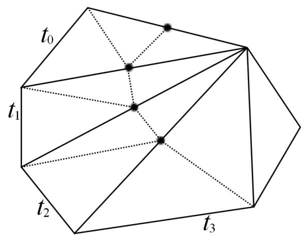

Figure 7: LEPP-midedge of t_0.

Rivara and collaborators ([37] and [49]) presented the LEPP-Delaunay midedge algorithm . It generalized and improved both previous longest-edge algorithms for the Rivara's refinement of general nonDelaunay Triangulations, and the longest-edge algorithm for the refinement of Delaunay meshes [38].

In the LEPP-Delaunay midedge algorithm, only considering local information associated to the terminal triangle that contains a constrained edge allows a real constrained Delaunay Triangulation. The constrained Delaunay Triangulation is the best approximation of the Delaunay Triangulation containing the set of given segments among its edges.

The LEPP-Delaunay midedge algorithm avoids the interaction with the entire set of constrained items. This algorithm is not a nested partition procedure because it changes the previously existing points. Moreover, it replaces previous triangles by Delaunay triangles due to the circumcircle test of DT(P). In addition, it suffers of a looping case for angle tolerance greater than $22°$. Namely, in certain cases, the triangles are not improved during the refinement. Nevertheless, it is interesting since it provides meshes with triangles which the smallest angle is greater than or equal to $\pi/6$, including along boundaries.

Hitschfeld and Rivara [29] introduced a automatic construction of nonobtuse triangles in boundary for LEPP-Delaunay Triangulations within control volume methods. Each 1-edge obtuse boundary triangle is eliminated by the Delaunay insertion of midedges.

Consider that α is the smallest angle of the triangle. In the case that $\alpha \geq 25.4°$, any isolated 1-edge obtuse triangle and isolated pairs of neighbour 1-edge obtuse triangles sharing their longest edge demand the insertion of only one point. When $\alpha \geq 15.4°$, the Delaunay insertion of at most three boundary/interface points eliminates any isolated 1-edge boundary triangle and isolated pairs of neighbour 1-edge boundary triangles sharing a longest edge. An obtuse angle in each isolated 2-edge boundary triangle having medium-size edge l and longest-size edge *L* over the boundary is eliminated by building an isosceles triangle of boundary edges of lengths *l*/2 followed by the Delaunay insertion of a finite number of points *N*, where $N \leq \frac{2.14}{sin(\alpha/2)}$.

A generalization of those approaches solves more complex patterns of obtuse triangles, i.e. chains of 2-edge constrained triangles forming a saw diagram and clusters of triangles that have boundary/interface edges sharing a common vertex [29]. Hitschfeld and collaborators [30] presented the LEPP algorithm for Delaunay mesh and its dual Voronoi Diagram, without obtuse angles opposite to the boundary and interfaces for semiconductor device simulation using Box-method Delaunay meshes.

Rivara and Calderon [36] presented the LEPP-Delaunay centroid algorithm. They proved that the centroid version of the LEPP-Delaunay algorithm produces triangulations both with average smallest angles greater than those obtained with the midedge version and with larger smallest edges without suffering from the looping case associated to the midedge method. In addition, the centroid version terminates for high-quality threshold angle, i.e. up to $\pi/5$. They also showed that the centroid version behaves better than the off-center algorithm for quality threshold angle larger than $25°$.

Because the finite-element conformity requirement, most of those previous articles describe algorithms that propagate the refinement in neighbors of the triangle marked to be refined and/or modify the points of the current mesh. As an example, Rivara and Inostroza [38] pointed out that numerical experiments performed with their 2D algorithm have shown that the number of points inserted by propagation is approximately $N^{1/2}$, where N is the number of points in the mesh.

If an algorithm modifies the positions of the refined-triangle points, the data-structure nodes that represent those triangles also have to be changed. A process that operates strict local changes (a nested mesh) is desirable. In [26] and [11], the authors described algorithms that perform the circumcircle test of DT(P) without locally destroying the current triangulation.

In [36], for constrained edges, in both the circumcenter and the off-center algorithm if a prospective point P to be inserted is inside the diameter circle of any constrained edge E, the midpoint of E is inserted instead of P. This implies that a strict Delaunay Triangulation is maintained. As a result, no angle lesser than $\pi/2$ appears in the triangulation.

5 Concluding remarks

Plaza and collaborators [34] provided several open problems related to their 7-triangle partition approaches. There is a lot of work related to 3D (for example, see [10]). In addition, the 3D review shall be provided.

The purpose of this article is to survey the approaches and not to evaluate them. Probably other schemes exist. However, such schemes may be either variations of the ones cited in this article or are not known to the author. However, the author hopes that this review and the references cited serve to consolidate the ideas, principles and schemes that constitute the state-of-art in this subject. Moreover, the author hopes that the list of references and descriptions to the large body of work on this issue can provide a useful starting point for one faced with the task of adaptively constructing a Delaunay Triangulation.

6 Acknowledgements

This work was supported by FAPEMIG under the project CEX-APQ-01198-10, year 2010.

References

[1] Babuzka, I. and Aziz, A. K. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13:214–226, 1976.

[2] Baker, T. J. Triangulations, mesh generation and point placement strategies. In Caughey, D., editor, *Computing the future. John Wiley*, pages 61–75, 1995.

[3] Ballard, D. H. and Brown, C. M. *Computer Vision*. Prentice Hall, Englewood Cliffs, NJ, 1982.

[4] Barth, T. J. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes Equations. In *Von Karman Institute for Fluid Dynamics Lecture Series, NASA Ames Research Center, 1994-05*, February 1995.

[5] Bern, M., Eppstein, D., and Gilbert, J. R. Provably good mesh generation. *Journal of Computer and System Sciences*, 48(3):384–409, June 1994.

[6] Borouchaki, H. and George, P. L. Aspects of 2-d Delaunay mesh generation. *International Journal for Numerical Methods in Engineering*, 40:1957–1975, 1997.

[7] Chew, L. P. Guaranteed-quality triangular meshes. Technical Report 983, Department of Computer Science, Cornell University, 1989.

[8] Chew, L. P. Constrained Delaunay triangulation. *Algorithmica*, 4:97–108, 1994.

[9] Clarkson, K. L. and Shor, P. W. Applications of random sampling in computational geometry ii. *Discrete & Computational Geometry*, 4(1):387–421, 1989.

[10] Danovaro, E., De Floriani, L., Magillo, P., Puppo, E., and Sobrero, D. Computer Graphics in Italy - Level-of-detail for data analysis and exploration: A historical overview and some new perspectives. *Computers & Graphics*, 30:334–344, 2006.

[11] De Floriani, L. Surface representation based on triangular grids. *The Visual Computer*, pages 27–48, 1987.

[12] De Floriani, L., Bussi, S., and Magillo, P. Triangle-based surface models. *Chapter 9 in Intelligent Systems and Robotics, Breach Science Publishers*, pages 340–373, 2000.

[13] De Floriani, L. and Puppo, E. A hierarchical triangle-based model for terrain description. In et al., A. U. F., editor, *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Berlin, Springer-Verlag*, pages 236–251, 1992.

[14] De Floriani, L. and Puppo, E. Extrating contour lines from a hierarchical surfase model. *Computer Graphics Forum (Proceedings Eurographics 93)*, 12(3):249–260, 1993.

[15] Delaunay, B. Sur la sphère vide. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7:793–800, 1934.

[16] Dwyer, R. A. A faster divide-and-conquer algorithm for constructing delaunay triangulations. *Algorithmica*, 2(2):137–151, 1987.

[17] Edelsbrunner, H. Triangulations and meshes in computational geometry. *Acta Numerica*, 9:133–213, 2000.

[18] Edelsbrunner, H. *Geometry and Topology for Mesh Generation. In: Cambridge monographs on applied and computational mathematics*. Cambridge University Press, New York, 2001.

[19] Edelsbrunner, H. and Guoy, D. Sink insertion for mesh improvement. In *Proceedings of the 17th ACM Symposium on Computational Geometry*, pages 115–123, 2001.

[20] Erten, H. and Üngör, A. Computing acute and non-obtuse triangulations. In *Canadian Conference on Computational Geometry (CCCG)*, pages 205–208, 2007.

[21] Erten, H. and Üngör, A. Triangulations with locally optimal Steiner points. In Belyaev, A. and Garland, M., editors, *Eurographics Symposium on Geometry Processing*, pages 143–152, 2007.

[22] Erten, H. and Üngör, A. Quality triangulations with locally optimal steiner points. *SIAM Journal of Scientfic Computing*, 31(3):2103–2130, 2009.

[23] Fortune, S. A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2(2):153–174, 1987.

[24] Fowler, R. and Little, J. Automatic extraction of irregular network digital terrain models. *ACM Computer Graphics (SIGGRAPH '79 Proceedings)*, 13(3):199–207, 1979.

[25] George, P. L. and Borouchaki, H. *Delaunay triangulation and meshing*. Hermes, 1998.

[26] Guibas, L. J. and Stolfi, J. Primitives for the manipulation of general subdivisions and the computation of Voronoi Diagrams. *ACM Transactions on Graphics*, 4(2):74–123, April 1985.

[27] Har-Peled, S. and Üngör, A. A time-optimal delaunay refinement algorithm in two dimensions. In *Proceedings of the 21st ACM Symposium on Computational Geometry, Pisa, Italy*, pages 228–236, 2005.

[28] Heckbert, P. S. and Garland, M. Survey of polygonal surface simplification algorithms. Technical report, Carnegie Mellon University - Department of Computer Science, 1 May 1997.

[29] Hitschfeld, N. and Rivara, M. C. Automatic construction of non-obtuse boundary and/or interface Delaunay triangulations for control volume methods. *International Journal for Numerical Methods in Engineering*, 55:803–816, 2002.

[30] Hitschfeld, N., Villablanca, L., Krause, J., and Rivara, M. C. Improving the quality of meshes for the simulation of semiconductor devices using LEPP-based algorithms. *International Journal for Numerical Methods in Engineering*, 58:333–347, 2003.

[31] Lee, D.-T. and Schachter, B. J. Two algorithms for constructing the delaunay triangulation. *International Journal of Computer and Information Science*, 9(3):219–242, 1980.

[32] Miller, G. L., Pav, S. E., and Walkington, N. J. When and why ruppert's algorithm works. In *Twelfth International Meshing Roundtable*, pages 91–102, September 2003.

[33] Mücke, E. P., Saias, I., and Zhu, B. Fast randomized point location without preprocessing in two- and three-dimensional delaunay triangulations. In *Proceedings of the Twelfth Annual Symposium on Computational Geometry*, May 1996.

[34] Plaza, A., Márquez, A., Moreno-González, A., and Suárez, J. P. Local refinement based on the 7-triangle longest-edge partition. *Mathematics and Computers in Simulation*, 79:2444–2457, 2009.

[35] Rivara, M. C. New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. *International Journal for Numerical Methods in Engineering*, 40:3313–3324, 1997.

[36] Rivara, M. C. and Calderon, C. LEPP terminal centroid method for quality triangulation. *Computer-Aided Design*, 42:58–66, 2010.

[37] Rivara, M. C., Hitschfeld, N., and Simpson, B. Terminal-edges Delaunay (small-angle based) algorithm for the quality triangulation problem. *Computer-Aided Design*, 33:263–277, 2001.

[38] Rivara, M. C. and Inostroza, P. Using longest-side bisection techniques for the automatic refinement of Delaunay triangulations. *International Journal for Numerical Methods in Engineering*, 40:581–597, 1997.

[39] Rivara, M. C. and Venere, M. Cost analysis of the longest-side (triangle biection) refinement algorithms for triangulations. *Engineering with Computers*, 12:224–234, 1996.

[40] Ruppert, J. A new and simple algorithm for quality 2-dimensional mesh generation. In *Proceedings of the Fourth ACM-SIAM Symposium on Discrete Algorithms*, pages 83–92, 1993.

[41] Ruppert, J. A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation. *Journal of Algorithms*, 18(3):548–585, May 1995.

[42] Scarlatos, L. and Pavlidis, T. Hierarchical triangulation using cartographic coherence. *CVGIP: Graphical Models and Image Processing*, 54(2):147–161, March 1992.

[43] Shamos, M. I. *Computational geometry*. PhD thesis, Yale University, New Haven, Conn., 1977.

[44] Shewchuk, J. R. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *First Workshop on Applied Computational Geometry. ACM*, pages 124–133, 1996.

[45] Shewchuk, J. R. Lecture notes on Delaunay mesh generation, 1999.

[46] Shewchuk, J. R. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22:21–74, 2002.

[47] Shewchuk, J. R. Ruppert's delaunay refinement algorithm. website, July 2005.

[48] Shewchuk, J. R. Triangle: A two-dimensional quality mesh generator and delaunay triangulator. website, July 2008.

[49] Simpson, B., Hitschfeld, N., and Rivara, M. C. Approximate shape quality mesh generation. *Engineering with Computers*, 17:287–298, 2001.

[50] Sleator, D. D. and Tarjan, R. E. Self-adjusting binary search trees. *Journal of the ACM*, 32(3):652–686, July 1985.

[51] Üngör, A. Off-centers: A new type of Steiner points for computing size-optimal quality-guaranteed Delaunay triangulations. In *Proceedings of the Latin American Symposium on Theoretical Informatics, Buenos Aires, Argentina*, pages 152–161, April 2004.

[52] Voronoi, G. Nouvelles applications des paramétres continus à la théorie des formes quadratiques. *Journal fur die Reine und Angewandte Mathematik*, 133:97–178, 1907.