On Computing New Classes of Optimal Triangulations with Angular Conditions *

Yang Dai
Department of Mathematical and Computing Sciences
Tokyo Institute of Technology
Naoki Katoh
Department of Architecture and Architectural Systems
Kyoto University

角度制約付き最適な三角形分割の計算について

戴 陽　東京工業大学 情報理工学研究科
加藤 直樹　京都大学 工学研究科

平面上の点集合について次のような二種類の角度制約付き最適な三角形分割を考える。(1) 三角形の最大角と最小角に制約を有する辺長の総和を最小にする三角形分割；(2) 三角形の最大角と最小角の比の総和を最小にする三角形分割。それぞれの最適な三角形分割について極小最適性を定義する。その定義にもとづいて最適な三角形分割の部分グラフを求める多項式時間アルゴリズムを与える。

Abstract

Given a planar point set, we consider two classes of optimal triangulations: (1) the minimum weight triangulation with angular constraints (constraints on the minimum angle and the maximum angle in the triangulation) and (2) the angular balanced triangulation which minimizes the sum of the ratios of the maximum angle to the minimum angle for each triangle. With the appropriate definition of local optimality for each class, we establish a simple unified method for the computation of the subgraphs of the optimal triangulations.

1 Introduction

Given a planar point set S, a triangulation of S is a maximal set of non-intersecting edges connecting points in S. Triangulating a point set has many applications in computational geometry and other related fields. Specifically, in numerical solutions for scientific and engineering applications, poorly shaped triangles can cause serious difficulty. Traditionally, triangulations which minimize the maximum angle, maximize the minimum angle, minimize the maximum edge length, and maximize the minimum hight are considered to be good. For example, if angles of triangles become too large, the discretization error in the finite element solution is increased and, if the angles become too small, the condition number of the element matrix is increased [1, 5, 13, 14]. Polynomial time algorithms have been developed in determining those triangulations [3, 10, 11, 18]. In computational geometry another important research object is to compute the minimum weight triangulation. The weight of a triangulation is defined to be the sum of the Euclidean lengths of the edges in the triangulation. Despite the intensive study made during the lase two decades, it remains unknown that whether the minimum weight triangulation problem is NP-complete or polynomially solvable.

In this paper we consider two new classes of optimal triangulations :

*This research is supported by the Grant-in-Aid of Ministry of Science, Culture and Education of Japan.

Problem (1) the minimum weight triangulation with the minimum angle in the triangulation not smaller than a given value α, and the maximum angle not greater than a given value γ;

Problem (2) the triangulation which minimizes the sum over all triangles of ratios defined by the values of the maximum angles to the minimum angles in the triangles.

Let $w(T)$ denote the weight of a triangulation T. Define for each triangle in the triangulation T a measure function $f(t) = \theta_{large}/\theta_{small}$, where θ_{large}(resp.θ_{small}) is the largest (resp. smallest) angle in the triangle. The problems defined above can be represented as follows.

Problem (1)

$$\begin{aligned} \min \quad & w(T) \\ \text{s.t.} \quad & \text{minimum angle} \geq \alpha, \\ & \text{maximum angle} \leq \gamma. \end{aligned}$$

Problem (2)

$$\min_T \sum_{t \in T} f(t).$$

If the value of α is zero and γ is equal to π, then Problem (1) is reduced to the minimum weight triangulation problem. If α is defined as the maximum value of the minimum angles among all possible triangulations, the solution of Problem (1) may give the Delaunay triangulation, although this case is not equivalent to the Delaunay triangulation problem. Therefore, Problem (1) contains the minimum weight triangulation as a special instance. We identify Problem (1) as the minimum weight triangulation problem with angular constraints and Problem (2) as the angular balanced triangulation problem. In Problem (1) we proposed somewhat more general criteria of the minimum weight triangulation and in Problem (2) a new criterion which is different from the minmax angle criterion. Although no evidences of applications of these new criteria have been found, we believe that they should be potentially useful, since the angular conditions on the angles in the minimum weight triangulation allows one to control the quality of the triangulation generated, and the sum of the ratios of the value over all triangles in the angular balanced triangulation contains more information than the minmax angle criterion does.

The main purpose of the paper is to provide solution methods for computing the optimal triangulations defined above. The difficulty of determining the optimal triangulation depends on the position of the points in the given set. If the points are vertices of a simple polygon then the problems are easily solvable. Actually the dynamic programming approach can be applied to both classes and provides polynomial time algorithms. For general point sets, the apparent difficult to compute the minimum weight triangulation problem means that it is unlikely that we can design a polynomial algorithm for the min-sum type problems at the moment. Moreover due to the angular conditions we can not expect the heuristic methods such as edge-flipping and greedy methods work well for these new classes. The edge-flipping method gets stuck in a local optimum even for the minimum weight triangulation problem without the angular constraints. The greedy algorithm may fail to provide a feasible solution for Problem (1)(see Figure 1). On the other hand, recent research has revealed promising ways to determine large subgraphs, namely, the β-skeleton [15, 16] and the LMT-skeleton [7, 8, 9] of the minimum weight triangulation. The experimental results show that these subgraphs are well connected for most of the point sets having relatively small sizes that are generated from uniform random distributions. Although Bose et al.[6] proved that the expected number of components in a subgraph is greater than one when n is larger than approximately 10^{51} for a random point set, the β-skeleton and the LMT-skeleton remain significant for the design of algorithms to compute the exact minimum weight triangulation of point sets having small sizes. When the number of connected components is small, a complete exact minimum weight triangulation can be produced by using the $O(n^3)$ dynamic algorithm [17] on each possible polygon which is reduced by the subgraph.

Unfortunately, the definition of the β-skeleton relies heavily on the distances of pairs of the points, it is not applicable to the new problems which involve angular conditions. However, the main idea of the LMT-skeleton for the minimum weight triangulation is the determination of the edges in every locally minimal triangulation. It suggests that there is room left to generalize the concept to the new classes of optimal triangulations through an appropriate definition of local optimality. This motivates us to design a generalized unifying method for the computation of subgraphs for other classes of optimal triangulations. Our new results are as follows.

- $O(n^3)$ time and $O(n^2)$ space algorithms for the problems in each class with the point set being a vertex set of a simple polygon.

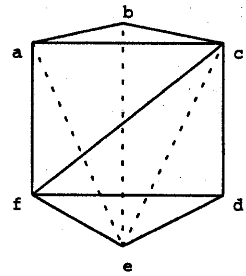

Figure 1: Suppose we need a minimum weight triangulations with the minimum angle to be greater than the angle $\angle bec$. Suppose also length(f,d) =length (a,c) < length(f,c) and the angle $\angle bac$ is the smallest one among all possible angles. If the greedy triangulation first chooses edge (f,d), then it generates the triangulation represented by solid lines. But it is not feasible for Problem (1). The dotted lines give a feasible triangulation.

- $O(n^4)$ time and $O(n^2)$ space algorithms for computing the subgraphs of the minimum weight triangulation with angular constraints and the angular balanced triangulation.

The organization of this paper is as follows. Section 2 gives polynomial time algorithm based on dynamic programming approach for computing the optimal triangulations defined above with the point set being a vertex set of a simple polygon. Section 3 presents the algorithm for the computation of the subgraph of the minimum weight triangulation with angular constraints. Section 4 introduces the algorithm for the determination of the subgraph of the angular balanced triangulation. Sections 5 states the conclusions.

2 Polynomial Time Algorithms

In this section we confine ourselves to the point set which is a vertex set of a simple polygon. We give polynomial time algorithms based on the dynamic programming approach for determining the minimum weight triangulation with angular constraints and the angular balanced triangulation.

In Bern and Eppstein [4] they discussed a class of optimal triangulation problems which admit efficient solutions. The class possesses so called decomposable measures which allow one to compute the measure of the entire triangulation quickly from the measures of two pieces of the triangulation, along with the knowledge of how the pieces are put together. The decomposable measures include the minimum (maximum) angle in the triangulation, the minimum (maximum) circumcircle of a triangle, the minimum (maximum) length of an edge in the triangulation, the minimum (maximum) area of a triangle, and the sum of edge lengths in the triangulation. They also presented polynomial time algorithms which use the dynamic programming approach attributed to Klincsek [17]. We will add the problems of determining the minimum weight triangulations with angular constraints and the angular balanced triangulation to the decomposable class and present polynomial time algorithms for solving these problems.

The Minimum Weight Triangulation with Angular Constraints:
To deal with the angular constraints we modify the algorithm of Klincsek [17] for the minimum weight triangulation. Denote the minimum weight triangulation of a point set S with respect to α and γ by $MWT(S,\alpha,\gamma)$. A triangle is defined admissible if it satisfies the angular conditions, otherwise it is inadmissible. Label the vertices $p_1, p_2, \ldots, p_n$ of the simple polygon in the clockwise order. We call edge p_ip_j $(i < j)$ diagonal if the line segment connecting p_i and p_j splits the polygon into two polygons whose union is the original polygon. For each diagonal edge p_ip_j, let $w(i,j)$ be the weight of the minimum weight triangulation of the polygon involving the points $p_i, p_{i+1}, \ldots, p_j$.
Algorithm $MWT(S,\alpha,\gamma)$:

Step1. For $k = 1, i = 1, 2, \ldots, n-1$ and $j = i + k$, let $w(i,j) = d(p_i, p_j)$, where $d(p_i, p_j)$ is the length of edge $p_i p_j$.

Step 2. Let $k = k + 1$. For $i = 1, 2 \ldots, n$ and $j = i + k \leq n$, if edge $p_i p_j$ is not diagonal and is not a boundary edge then let $w(i,j) = +\infty$. Otherwise test for each $i < m < j$ whether the triangle $p_i p_m p_j$ is admissible.

Let $M = \{m \mid i < m < j,\ p_i p_m p_j$ is admissible and both edge $p_i p_m$ and edge $p_m p_j$ are diagonal$\}$. Compute

$$w(i,j) = \begin{cases} d(p_i, p_j) + \min_{m \in M}\{w(i,m) + w(m,j)\} & \text{for } M \neq \emptyset \\ +\infty & \text{otherwise.} \end{cases} \quad (1)$$

For each pair (i,j) such that $w(i,j) < \infty$, let $m^*(i,j)$ be the index where the minimum $w(i,j)$ in (1) is achieved.

Step 3. If $k < n - 1$, go to Step 2. Otherwise $w(1,n)$ is the minimum weight.

Step 4. If $w(1,n) < \infty$, then backtrace along the pointers m^* to determine the edges of the minimum weight triangulation. Otherwise no triangulation satisfying the angular conditions exists.

The Angular Balanced Triangulation:
Let t be an arbitrary triangle in some triangulation T. Recall that the measure function $f(t)$ of the triangle t is defined by $f(t) = \theta_{\text{large}}/\theta_{\text{small}}$. For any two triangles t and t', let $f(t,t') = f(t) + f(t')$. The sum of the ratios of the triangulation T, denoted by $f(T)$, is defined as $f(T) = \sum_{t \in T} f(t)$. Therefore, the angular balanced triangulation is a triangulation T that minimizes the value $f(T)$ over all triangulations.

We denote by $P(i,j)$ the polygon formed by points $p_i, p_{i+1}, \ldots, p_j$. Let $F(i,j)$ be the minimum value of $f(T)$ over all triangulations T of $P(i,j)$. Define $F(i,j) = f(i,j) = +\infty$ for each non-diagonal edge $p_i p_j$. We compute $F(1,n)$ by the dynamic programming method. Suppose that an arbitrary diagonal edge $\{a,b\}$ splits $P(i,j)$ into two polygons P_1 and P_2. Let T_1 and T_2 be the triangulations of P_1 and P_2, respectively. We define a function g as follows.

$$g(f(T_1), f(T_2), a, b) = f(T_1) + f(T_2) = f(T).$$

If edge $\{a,b\}$ is not diagonal and is on the boundary of the polygon $P(1,n)$, then we define in this case

$$g(f(T), f(\{a,b\}), a, b) = f(T).$$

Note that in any triangulation of $P(i,j)$, $p_i p_j$ must be a side of a triangle, say $p_i p_j p_m$, with $i < m < j$. We can compute the value of the angular balanced triangulation of $P(i,j)$ by trying all choices of m. Therefore the algorithm can be obtained by replacing $w(i,j)$ with $F(i,j)$ and **Step 2** with the following **Step 2'** in **Algorithm** $MWT(S, \alpha, \gamma)$.

Sept 2'. let $k = k + 1$. For $i = 1, 2 \ldots, n$ and $j = i + k \leq n$, Compute

$$F(i,j) = \min_{i<m<j} g(g(f(p_i p_j p_m), F(i,m), p_i, p_m), F(m,j), p_m, p_j)) \quad (2)$$

For each pair (i,j) let $m^*(i,j)$ be the index where $F(i,j)$ in (2) is achieved.

Note that test whether a pair (i,j) forms a diagonal takes $O(n)$ time. Therefore following a similar analysis as in [4, 17] one can show that the both of the algorithms take $O(n^3)$ time and $O(n^2)$ space.

3 The Subgraph of the Minimum Weight Triangulation with Angular Constraints

In this section we present the algorithm for computing the subgraph of the minimum weight triangulation with angular constraints for point set in general position. Designate the set of all possible edges connecting two points in S by $E(S)$. We assume that the values of α and γ are given so that there are always triangulations satisfying the angular constraints. In order to determine the minimum weight triangulation $MWT(S,\alpha,\gamma)$, we only need to consider triangulations which contain no inadmissible triangles. For the entirety of our discussion in this section, the term triangulation will mean a triangulation which consists solely of admissible triangles.

Let e be an edge in an arbitrary triangulation. If e is not on the boundary of the convex hull of the set S, then there exist two admissible triangles t_1 and t_2 such that $t_1 \cap t_2 = e$. If the quadrilateral $t_1 \cup t_2$ is convex, then it contains another diagonal, e'. Denote by t_1' and t_2' as the two triangles formed by connecting the edge e'. The edge e is defined to be locally minimal with respect to (α,γ) if either one of the following two cases holds.

case 1: $t_1 \cup t_2$ is not convex.

case 2: $t_1 \cup t_2$ is convex and either (i) $|e| \leq |e'|$, or (ii) at least one of the triangles t_1' and t_2' is inadmissible.

A triangulation is called locally minimal with respect to (α,γ) if each edge e in the triangulation is locally minimal with respect to the two triangles containing e and (α,γ). From the definition, it follows that $MWT(S,\alpha,\gamma)$ is a locally minimal triangulation with respect to (α,γ). The intersection of all locally minimal triangulations must be a subgraph of $MWT(S,\alpha,\gamma)$.

We define a triangle as empty if it contains no points of S except the vertices. For each edge and empty triangle if they are known not to be contained in any locally minimal triangulation we call them dead. Therefore all inadmissible triangles are dead initially. When each edge is examined, its status will be determined as active, inactive, or dead as follows. Let $\mathcal{T}$ be the set of pairs $\{axb, ayb\}$ of empty active triangles, one from each side of edge ab such that $axb \cap ayb = ab$. The edge ab is labeled active if it lies on the boundary of the convex hull of S, or if there exists $\{axb, ayb\} \in \mathcal{T}$ such that $ab \cap xy \neq \emptyset$ and (i) $|ab| \leq |xy|$, or (ii) at least one of the triangles xay and xby is inadmissible.

Suppose that ab is not labeled active. Then, if $\mathcal{T} = \emptyset$, or $ab \cap xy \neq \emptyset$ for all $\{axb, ayb\} \in \mathcal{T}$, we label ab dead. Otherwise, we label ab inactive. If an edge ab becomes inactive or dead, we label some of the admissible triangles bounding ab as dead. More precisely, define the set $\mathcal{A}$ as the collection of the empty admissible active triangles axb which satisfy:

> $axb \cap ayb = ab$, $ab \cap xy \neq \emptyset$ for all empty admissible active triangles ayb such that x, y are on different sides of ab, respectively.

We label all triangles in the set $\mathcal{A}$ dead. The following two lemmas guarantee the correctness of the algorithm.

Lemma 1 *If an empty admissible triangle t is labeled dead, then $t \notin MWT(S,\alpha,\gamma)$.*

Lemma 2 *If $ab \notin MWT(S,\alpha,\gamma)$, then ab intersects some active edge.*

We present below an algorithm which computes some of the edges in the intersection of all locally minimal triangulations with respect to (α,γ). When the algorithm terminates, it produces a set of edges. We name this set the LMT-skeleton of the minimum weight triangulation with angular constraints and denote it by LMT-skeleton(S,α,γ). The algorithm for computing the subgraph of MWT(S,α,γ) is given below.

Three edge sets are used in the algorithm; they are *candEdges*, *edgeIn* and *deadEdges*. All edges in $E(S)$ are initially active. We note additionally that (1) all edges in $E(S)$, except the convex hull edges, are in *candEdges*, (2) *edgeIn* contains the convex hull edges and (3) *deadEdges* is empty.

The Algorithm LMT(S,α,γ):
Input: point set S.
Output: edge set *edgeIn*.

Step 0. Set all edges of *candEdges* unexamined.

step 1. If there are no unexamined edges, go to **Step 4**. Otherwise choose an unexamined edge $e \in$*candEdges*, check all empty triangles on both sides of e. If they are all inadmissible then delete e from *candEdges* and move it to *deadEdges*. Otherwise,

Step 2. Find all combinations of empty admissible triangles t_i and t_j on the two sides of e such that t_i and t_j are not bordered by an edge in *deadEdges*.

Step 3. For each combination of t_i and t_j, test if e is locally minimal with respect to t_i, t_j and (α, γ). If e is not locally minimal to any such pair t_i to t_j, then move e to *deadEdges*. Otherwise, mark e active or inactive according to the definitions. Go to **Step 1.**

Step 4. For each edge marked active or inactive, if it intersects no other active edges then move it to *edgeIn*.

The only difference of the Algorithm LMT(S, α, γ) with the algorithm in [7] is the admissible test for each empty triangle. This can be tested in $O(1)$ time. Therefore, following a similar discussion on the complexity analysis [7], the Algorithm LMT(S, α, γ) can be run in $O(n^4)$ time and $O(n^2)$ space for uniformly distributed point sets.

Figures 2-5 show results of LMT-Skeletons LMT(S, α, γ) by testing a set of 64 points with the angle α changing between 0 and the minimum angle α_D of the Delaunay triangulation, i.e., $0 = \alpha_1 < \alpha_2 < \alpha_3 < \alpha_4 < \alpha_D$. We set the angle $\gamma = \pi$.

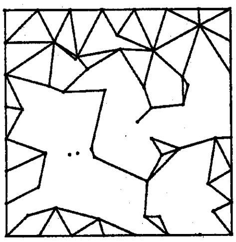

Figure 2: LMT-skeleton(S, α_1, γ)

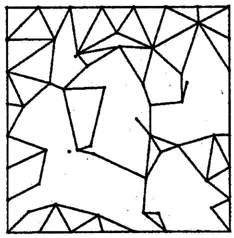

Figure 3: LMT-skeleton(S, α_2, γ)

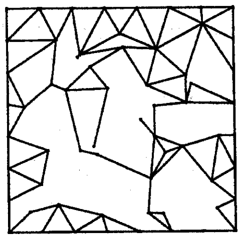

Figure 4: LMT-skeleton(S, α_3, γ)

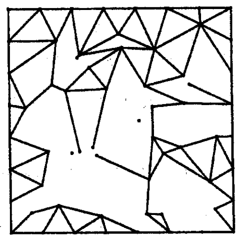

Figure 5: LMT-skeleton(S, α_4, γ)

4 The Subgraph of the Angular Balanced Triangulation

Denote the angular balanced triangulation of a general position point set S by ABT(S). Initially, we define the concept the local optimality of the angular balanced triangulation, and subsequently give an algorithm which can be used to determine some of the edges in every locally angular balanced triangulation.

Let e be an arbitrary edge in some triangulation but not on the convex hull. As discussed in Section 3, there exist two triangles t_1 and t_2 such that $t_1 \cap t_2 = e$ and a corresponding pair of triangles t'_1 and t'_2 such that $t'_1 \cap t'_2 = e'$. The edge e is defined to be locally angular balanced if $t_1 \cup t_2$ is not convex or if $t_1 \cup t_2$ is convex and $f(t_1, t_2) \leq f(t'_1, t'_2)$. A triangulation is called a locally angular balanced triangulation (LABT) if each of its edge is locally angular balanced. Obviously, the angular balanced triangulation must be locally angular balanced; otherwise the exchange of the diagonals in a convex quadrilateral containing a non-balanced edge will reduce the sum of the values.

Naturally, a concept similar to the LMT-skeleton for the minimum weight triangulation can be considered. Correspondingly, we name the subgraph which contains a set of edges that must be in every locally angular balanced triangulation the LABT-skeleton and denote it by LABT(S). Our algorithm for the computation of the LABT-skeleton is presented below. In the algorithm an edge has two status. More precisely, if an edge is not in any locally angular balanced triangulation then it is dead. Otherwise it is active.

The Algorithm LBAT(S):
Input : point set S.
Output: edge set *EdgeIn*.

Step 1. Set *candEdge*=all edges in $E(S)$ except edges on the convex hull of S; *DeadEdge*=$\emptyset$; *edgeIn*=all edges on the convex hull.

Step 2. For each edge $e \in$ *candEdge*

- **(a)** Test each combination of empty triangles t_i and t_j that are on different sides of e and neighboring of e to see if e is locally angular balanced with respect to t_i and t_j. If e is not locally angular balanced to any such pair, then remove e from *candEdge* to *DeadEdge*. Otherwise,
- **(b)** If e intersects no other edges in *candEdge* or *EdgeIn*, then add e to *edgeIn*.

The following three lemmas establish the correctness of the algorithms.

Lemma 3 *For any point set S, an edge in every locally angular balanced triangulation of S must be an edge of LABT(S), and an edge not in any locally angular balanced triangulation of S must not be an edge of LABT(S).*

Lemma 4 *Let e be in any locally angular balanced triangulation, and let $T_L(e)$ (resp. $T_R(e)$) be the set of empty triangles that border e on left (resp. right) side. Then there exist some $t_1 \in T_L(e)$ and $t_2 \in T_R(e)$ such that e is locally angular balanced with respect to the quadrilateral formed by t_1 and t_2.*

Lemma 5 *Let e be in some locally angular balanced triangulation. If e does not intersect any other edges in $E(S)$, then e is in all locally angular balanced triangulations.*

Computing the empty triangles for each edge requires $O(n \log n)$ time by using the method of [7], so totally $O(n^3 \log n)$ for all the edges. We can also preprocess the data in $O(n^2)$ time and $O(n^2)$ space by the algorithm in [12] so that all empty triangles sharing an edge can be computed in linear time. The loop in **Step 2** iterates $O(n^2)$ times, once for each candidate edge. To test if an edge is locally angular balanced can be done in $O(1)$ time. Since e might have a linear number of triangles on each side, we may test $O(n^2)$ combinations of adjacent triangles. In Step (b) We test an edge against at most $O(n^2)$ other edges. Thus, the Algorithm LABT(S) takes $O(n^4)$ time and $O(n^2)$ space.

5 Conclusions

Optimal triangulations can be constructed on the basis of certain locally established criteria. This work has considered two classes, one involving a minimum weight triangulation with angular constraints and another triangulation based on the minimization of the ratio of corresponding triangular angles. It has been found that the identification of the appropriate definition of local optimality for each class leads to a simple unified method for the computation of the subgraphs of the optimum triangulations. It is worth noticing that this unified method also works for determining the subgraph of other optimal triangulations with min-sum type quality measurement. For example, if the measure function f for each triangle is defined as the ratio of the length of the longest edge to that of the shortest edge, the triangulation with the minimum value is the one that balances the lengths of the edges. We hope that further investigation on the minimum wright triangulation with angular constraints provides us insight into the design of algorithms for solving the minimum weight triangulation.
Acknowledgment: We thank Mr. Manabu Sugai and Professor Kokichi Sugihara and Professor Masao Iri [19] for sharing us their programs for the implementation of our algorithms.

References

[1] I. Babuska and A. Aziz, "On the angle condition in the finite element method", *SIAM Journal on Numerical Analysis* 13, pp.214-226, 1976.

[2] P. Belleville, M. Keil, M. McAllister, and J. Snoeyink, "On computing edges that are in all minimum-weight triangulations", Video Presentation, *Proceedings of the 12th Annual ACM Symposium on Computational Geometry*, pp. V7-V8, 1996.

[3] M. Bern, H. Edelsbrunner, D. Eppstein, S. Mitchell, and T. S. Tan, "Edge insertion for optimal triangulations", *Discrete Comput. Geom.*, 10, pp.47-65, 1993.

[4] M. Bern and D. Eppstein, "Mesh generation and optimal triangulation", in D.-Z. Du eds. *Computing in Euclidean Geometry*, World Scientific Publishing, pp.23-90, 1992.

[5] H. Borouchaki, F. Hecht, E. Saltel and P.L. George, "Reasonably efficient Delaunay based mesh generator in 3 dimensions", *Proceedings of the 5th International Mesh Roundtable*, pp. 87-100, 1996.

[6] P. Bose, L. Devroye and W. Evans, "Diamonds are not a minimum weight triangulation's best friend", Technical Report 96-01, Dept. of C.S., Univ. of Brit. Columbia, 1996.

[7] S.-W. Cheng, N. Katoh and M. Sugai, "A study of the LMT-Skeleton", *Proceedings of ISAAC'96*, Lecture Notes in Computing Science 1178, pp.339-359, 1996.

[8] S.-W. Cheng and Y. Xu, "Approaching the largest $\beta-$skeleton within a minimum weight triangulation", *Proceedings of the 12th Annual ACM Symposium on Computational Geometry*, 1996.

[9] M. T. Dickerson and M. H. Montague, "The exact minimum weight triangulation", *Proceedings of the 12th Annual ACM Symposium on Computational Geometry*, 1996.

[10] H. Edelsbrunner and T. S. Tan, "A quadratic time algorithm for the minmax length triangulation ", *SIAM J. Comput.* 22, pp. 527-551, 1993.

[11] H. Edelsbrunner, T. S. Tan and R. Waupotitsch, "A $O(n \log n)$ time algorithm for the minmax angle triangulation", *SIAM J. Sci. Statist. Comput.* 13, pp. 994-1008, 1992.

[12] D. Eppstein, M. Overmars, G. Rote and G. Woeginger, "Finding Minimum Area k-gons", *Discrete Comput. Geom.* 7, pp. 45-58, 1992.

[13] L. Freitag, M. Jones and P. Plassmann, "An efficient parallel algorithm for mesh smoothing", *Proceedings of the 5th International Mesh Roundtable*, 1996.

[14] I. Fried, "Condition of finite element matrices generated from nonuniform meshes", *AIAA Journal* 10, pp.219-221, 1972.

[15] J. M. Keil, "Computing a subgraph of the minimum weight triangulation", *Computational Geometry: Theory and Applications*, Vol. 4, pp.13-26, 1994.

[16] D. Kirkpatrick and J. Radke, "A framework for computational morphology", in G. Toussaint ed, *Computational Geometry* , pp. 217-248, Elsevier Science Publishers, 1985.

[17] G. Klincsek, "Minimum triangulations of polygonal domains", *Annual Discrete Mathematics* 9, pp.121-123, 1980.

[18] R. Sibson, "Locally equiangular triangulations", *Comput. J*, 21, pp.243-245, 1978.

[19] K. Sugihara and M. Iri, "VORONOI2 reference manual, Research Memorandum RMI-89-04, Department of Mathematical Engineering and Information Physics, Faculty of Engineering, University of Tokyo, 1993.