Properties and Algorithms for Constrained Delaunay Triangulations

by

Teofilo Gonzalez and Mohammadreza Razzazi‡
Department of Computer Science
University of California
Santa Barbara, CA 93106

EXTENDED ABSTRACT:

We present an O(n log n) algorithm to construct a constrained Delaunay triangulation (*CDT*) for a simple polygon with interior points and holes. The main difference between our algorithm and the previous algorithms for the *CDT* problem is that our algorithm reduces the problem to a set of line intersection problems plus finding Delaunay triangulations (*DT*) on several simple polygons plus finding Delaunay triangulations on several sets of points. Our algorithm is based on partitions that exploit useful properties of constrained Delaunay triangulations.

Let Q be a set of points in the plane. An *edge for Q (or simply an edge)* is a line segment that joins two points in Q and a *triangulation* for Q is a maximal set of edges for Q no two of which cross. A *Delaunay Triangulation (DT)* for Q is any triangulation for Q with the additional property that for each edge e in it there exists a circle C that passes through the endpoints of e and there is no vertex of Q in the interior of C. In a Delaunay triangulation each edge is called a *Delaunay edge* and each triangle is called a *Delaunay triangle*. The dual of the Delaunay triangulation problem is the problem of constructing a *Voronoi Diagram* [SH]. A Voronoi Diagram for a set of points $Q = \{q_1, q_2, ... q_n\}$ in the plane consists of a set of edges that partition the plane into a set of regions $R = \{r_1, r_2, ..., r_n\}$ with the property that for each i, $q_i \in r_i$ and each point in r_i is not farther from q_i than from q_j, for all $j \neq i$. Shamos and Hoey [SH] developed an O(n log n) algorithm to construct the Voronoi diagram for Q and showed that a Delaunay triangulation for Q may be obtained from it in O(n) time. They also showed that any "decision tree" type algorithm must take at least Ω(n log n) time to solve either of the two problems. Lee and Schacter [LS] developed an algorithm to obtain Delaunay triangulations in O(n log n) time, without constructing the Voronoi diagram. Generalization of these two problems have been extensively studied (see for example [WS], [Cw], [C], [S] and [PS]).

In this paper we study a generalization of the Delaunay triangulation which is known as the *Constrained Delaunay Triangulation (CDT)* problem. Let P be a simple polygon, and let H be a set of pairwise disjoint simple polygons, called *holes*, defined inside P. The edges of P are called *boundary edges* and the edges of the holes in H are called *hole edges*. Let D be a set of points inside IP -RH, where IP is the region inside P and RH is the region

‡ Current address: Computer Engineering and Science Department, Case Western Reserve University, Cleveland, Ohio 44106.

inside and the boundary of H. We define $V(A)$, where A is a set of pairwise disjoint polygons, as the union of the set of vertices in each polygon in A. Let $Q = D \cup V(P) \cup V(H)$ and let n be the cardinality of Q. A *Constrained Delaunay Triangulation (CDT) for Q restricted by P and H* is a set of edges for Q that satisfy the circle property and when added to IP-RH partitions the region IP-RH into triangles. An edge e is said to satisfy the *circle property* if there is a circle C that passes through the endpoints of e and which does not include inside it any other point in Q that is visible (when considering boundary and hole edges as obstacles) from both of the endpoints of e. It is simple to see that the constrained Delaunay triangulation problem reduces to the Delaunay triangulation for Q when P is a convex polygon and there are no holes. When each hole degenerates into a single line segment, we refer to the *CDT* problem as the *CDT′* problem. It is simple to see that the *CDT* and the *CDT′* are computationally equivalent problems. A brute force algorithm for the *CDT* problem was developed by Nielson and Franke [NF]. For the *CDT′* problem, Lee and Lin [LL] presented an $O(n^2)$ algorithm and Chew [Cw] developed a divide-and-conquer algorithm that takes optimal time, i.e, $O(n \log n)$ time. Wang and Shubert [WS] introduced the notion of Bounded Voronoi diagram and showed how to find the Bounded Voronoi diagram and the *CDT′* in $O(n \log n)$ time. Jung ([J1], [J2]) adapted this algorithm to find the *CDT′* directly. As noted in [J1], the performance of the algorithm degrades as the number of hole edges increases. Seidel [S] developed the notion of an Extended Voronoi Diagram (EVD) and showed that it is the dual of the *CDT′* problem. Based on this concept and on Fortune's [F] sweep line technique, Seidel [S] developed an elegant $O(n \log n)$ algorithm for constructing the extended Voronoi diagram and by duality the *CDT′*. All of these algorithms can be easily adapted to solve the *CDT* problem.

Let us now explain these algorithms in more detail in order to compare them with our algorithm. Chew's [Cw] algorithm begins by sorting the set of points Q along their x-coordinate values, and a box that covers all the points is defined. The box is divided into vertical strips each containing exactly one point, since it is assumed that all points have distinct x-coordinate values these strips exist. The *CDT′* is constructed on each strip and then the triangulations of adjacent strips are combined until the *CDT′* of the entire problem is obtained. The process takes $O(n \log n)$ time. Jung's method ([J1] and [J2]), which is exactly the dual of Wang and Shubert's method [WS] is different. First, the *DT* of the set of points Q is constructed via Lee and Schacter's algorithm [LS]. If all the hole edges overlap with the Delaunay edges, then the *CDT′* is just the *DT* and the algorithm terminates. Otherwise, the edges in the *DT* that intersect hole edges are deleted. Because of this, some regions need to be retriangulated. The resulting area which is not triangulated is partitioned into polygons, called *difference polygons*, by introducing auxiliary edges in such a way that each difference polygon contains exactly one hole edge. Each difference polygon is triangulated by the $O(n \log n)$ algorithm developed by Lee and Lin [LL] that is based on Chazelle's [Ch] divide-and-conquer partitioning rule for polygons. Once each of the difference polygons is triangulated it is combined with the previous Delaunay edges. The auxiliary edges are removed in certain order and each time several adjacent difference polygons are merged. The claim is that the algorithm has an $O(n \log n)$ time complexity bound. Seidel's [S] method is based on the concept of Extended Voronoi Diagram and constructs the EVD using Fortune's sweep line technique. The *CDT′* is obtained by duality. The time complexity bound for Seidel's algorithm is also $O(n \log n)$.

Our method is different, though it is closest to the one given by Jung ([J1] and [J2]), which is based on Wang and Shubert's method [WS]. The first step consists of finding the *DT* of all the points in Q. If the hole edges overlap with the Delaunay edges, we have the *CDT* and the algorithm terminates. Otherwise we proceed as follows. Instead of deleting a Delaunay edge that intersects a hole edge, as in Jung's method, we replace the edge $(a\,b)$ by edges $a\,i_a$ and $i_b\,b$ where i_a (i_b) is the closest point to vertex a (b) on edge $a\,b$ that intersects a hole edge. As a result of this operation a new set of points is introduced. These points are called *E_points*, whereas the original points are referred to as *S_points*. Then we delete all edges outside the boundary of P or inside the holes. We now take one hole edge and delete all the *E_points* on it as well as all the edges incident to these *E_points*. The resulting polygon is called the *cut of the line*. We modify the cut so that it satisfies some additional properties and call it the *mcut of the line*. This *mcut* is triangulated by the O(n log n) algorithm developed by Lee and Lin [LL] that is based on Chazelle's [C] divide and conquer partitioning rule for polygons. We add these edges to the previously introduced edges. This procedure is applied to each hole edge with *E_points* on it. The resulting edges form a *CDT*.

The main difference between our procedure and Jung's procedure is that in our procedure each edge which is added inside the polygon (*mcut*) satisfies the circle property. For the difference polygons, this is not necessarily true, i.e., the property is satisfied inside the difference polygon but not necessarily outside it. This is why he needs to merge adjacent difference polygons after deleting auxiliary edges. Because of this property we say that *mcuts* are more natural than difference polygons. The implication of this property of *mcuts* is that our algorithm reduces the *CDT* problem to a set of line intersection problems plus finding Delaunay triangulations on several simple polygons plus finding Delaunay triangulations on several sets of points. This is the first step in reducing the *CDT* problem to finding a *DT* on sets of points plus solving other simple problems. This is important since such a result would imply the "direct equivalence" of the *DT* and *CDT* problems. The interesting point is that some of the probabilistic analyses for the DT problem may directly translate to the *CDT* problem. Also, our algorithm is based on partitions that exploit useful properties of constrained Delaunay triangulations. This is similar in nature to the results of Guibas and Stolfi [GS] for finding a DT for a set of line segments.

For brevity we cannot state the algorithm formally. Neither its correctness proof nor its time complexity bound proof are simple. They involve proving some geometrical properties of triangulations, and circles, as well as establishing some relations to account for the steps taken by the algorithm. Our main results are given in the following theorem which for brevity we do not prove.

Theorem 1: Our algorithm, which we call FIND_CDT, constructs a *CDT* for Q restricted by P and H in O(n log n) time.

Proof: For brevity the proof is omitted.

□

We outlined an O(n log n) algorithm to construct a constrained Delaunay triangulation for a simple polygon with interior points and holes. The main difference between our algorithm and the previous algorithms for this problem is that our algorithm reduces the problem to a set of line intersection problems plus finding Delaunay triangulations on several simple polygons plus finding Delaunay triangulations on several sets of points. This is the first step in reducing the *CDT* problem to finding a *DT* on sets of points plus solving other simple problems. This is important since such a result would imply the "direct equivalence" of the *DT* and *CDT* problems. The interesting point is that some of the probabilistic analyses for the DT problem would directly translate to the *CDT* problem. Also, our algorithm is based on partitions that exploit useful properties of constrained Delaunay triangulations.

REFERENCES

[C] Chazelle, B. M., "A Theorem on Polygon cuting with Applications," Proceedings of the 23rd IEEE Annual Symposium on Foundations of Computer Science, 1982, 339 - 349.

[Cw] Chew L., "constrained Delaunay Triangulations," Proceedings of the 3rd Annual Symposium on Computational Geometry, 1987, pp. 215 - 222.

[J1] Jung, D., "An Optimal Algorithm for Constrained Delaunay Triangulations," Proceedings of the 1988 Annual Allerton Conference on Communications, Control and Computing, pp. 83 - 84, October 1988.

[J2] Jung, D., "On Constructing the Constrained Delaunay Triangulation," Technical Report 100, Electrical, Computer and Systems Engineering Department, Rensselaer Polytechnic Institute, 1988.

[F] Fortune, S. J., "A Sweepline Algorithm for Voronoi Diagrams," *Algorithmica, 1987.*

[GS] Guibas, L. and Stolfi, J., "Primitives for the Manipulation of General Subdivision and the Computation of Voronoi Diagrams," *ACM Transactions on Graphics,* Vol. 4, No 2, pp 74 - 123, April 1985.

[L] Lawson, C. L., "Software for C_1 Surface Interpolation," in Mathematical Software III (J. R. Rice, Ed.), Academic Press, New York, 1977, pp. 161 - 194.

[LS] Lee D. T. and B. J. Schacter, "Two algorithms for Constructing Delaunay Triangulations," *International Journal of Computer and Information Sciences,* 9, 3, (1980), pp 219 - 242.

[LL] Lee D. T. and A. K. Lin, "Generalized Delaunay Triangulation for Planar Graphs," *Discrete Computational Geometry,* 1, (1986), 201 - 217.

[NF] Nielson G. N. and Richard Franke, "Surface Construction Based Upon Triangulations," *Surfaces in CAGD,* North-Holland Publishing Co., 1983, 163 - 177.

[PS] Preparata, F. P. and M. I. Shamos, *"Computational Geometry,"* Springer-Verlang, 1985.

[S] Seidel, R., "Constrained Delaunay Triangulations and Voronoi Diagrams with Obstacles," Computer Science Division, UC Berkeley, June 1989.

[SH] Shamos, M. I. and D. Hoey, "Closest Point Problems," Proceedings of the 16th Symposium on Foundations of Computer Science, October 1975, 151 - 162.

[WS] Wang C. and L. Schubert, "An Optimal Algorithm for Constructing the Delaunay Triangulation of a Set of Line Segments," Proceedings of the 3rd Computational Geometry Conference, 1987, pp. 223 - 232.