FAST ALGORITHMS FOR BOUNDED VORONOI DIAGRAMS OF RESTRICTED POLYGONS

Andrzej Lingas
Department of Computer Science
Lund University, Box 118, 22100 Lund, Sweden

Abstract: By using a divide-and-conquer strategy, we can naturally extend the linear-time bound on the construction of Voronoi diagram of a convex polygon to include an $O(n \log r)$-time bound on the construction of a bounded Voronoi diagram of a triangulated polygon with r reflex angles. A partition of a simple polygon into simple polygons is an m-approximation of the bounded Voronoi diagram of P if none of the regions in the diagram intersects more than m edges in the partition. Given a convex m-approximation of the bounded Voronoi diagram of P, we can construct the bounded Voronoi diagram of P in time $O(nm^2)$. We also discuss cases of restricted convex m-approximations of P where it is possible to construct the diagram for P in time $O(nm)$.

1. Introduction

Planar point sets and simple polygons are special cases of the so called planar straight-line graphs (PSLG for short) [PS]. A PSLG G is a pair (V, E) such that V is a set of points in the plane and E is a set of non-intersecting, open straight-line segments whose endpoints are in V. The points in V are called vertices of G, whereas the segments in E are called edges of G. If G is a simple cycle, it is called a (simple) *polygon*. If G has no edges, it is a *planar point set*. The so called *bounded Voronoi diagram* of a PSLG $G = (V, E)$ (called Voronoi diagram with barries in [L89, LL90]) can be defined as follows [L89,WS].
For $v \in V$, the region $P(v)$ consists of all points p in the plane for which the shortest, open straight-line segment between p and a vertex of G that does not intersect any edge of G is (p, v). The minimal set of straight-line segments and half-lines that complements G to the partition of the plane into the regions $P(v)$, $v \in V$, is called the bounded Voronoi diagram of G ($Vorb(G)$ for short). The maximal straight-line segments or half-lines on the boundaries of the regions $P(v)$, $v \in V$, that do not overlap with edges of G, are called edges of $Vorb(G)$. The endpoints of edges of $Vorb(G)$ are called vertices of $Vorb(G)$.
Note that if G is a planar point set then $Vorb(G)$ is just the standard Voronoi diagram of G [PS]. We shall denote the standard Voronoi diagram of a planar point set S by $Vor(S)$. For a PSLG G, $Vorb(G)$ has several properties analogous to those of standard Voronoi diagrams (comp. [L89] with [PS]). Every edge of $Vorb(G)$ is a continuous part of the perpendicular bisector of a pair of vertices of G from which the edge is visible. Next, every vertex of $Vorb(G)$ is the common intersection of at least three edges of $Vorb(G)$ or residues inside an edge of G. For all $v \in V$, $P(v)$ is a collection of polygonal sub-regions separated by the edges incident to v. There are at most $3n - 6$ edges in $Vorb(G)$ (see [L89]). For convention, given a simple polygon P, we shall denote by $Vorb(P)$ the bounded Voronoi diagram of P within P throughout the paper.
In [LL87], Lee and Lin presented an $O(n \log n)$-time divide-and-conquer algorithm for the so called *generalized Delaunay triangulation* of a simple polygon which is the straight-line dual (see [PS]) of the bounded Voronoi diagram of the polygon [L89,WS]. Later, Wang and Schubert showed that the bounded Voronoi diagram of an arbitrary PSLG can be constructed also in time which is asymptotically optimal [WS]. On the other hand, it is not known whether the $n \log n$ upper time-bound for simple polygons is asymptotically optimal. Aggarwal put the above question on his list of important open problems in computational geometry [A].
As the straight-line dual of the bounded Voronoi diagram of a simple polygon P can be easily completed to a full triangulation of P, any fast algorithm for $Vorb(P)$ yields a fast

algorithm for a triangulation of P. For this reason, it is doubtful whether one could derive a sub-$n \log\log n$ algorithm for $Vorb(P)$ presently (see [TV]). On the other hand, Aggarwal, Guibas, Saxe and Shor presented a linear-time algorithm for the Voronoi diagram of a set of points which form a convex polygon [AGSS]. They result can be stated equivalently as a linear-time algorithm for the bounded Voronoi diagram of a convex polygon.
In this paper, we partially fill the gap between the linear-time upper bound for convex polygons and the $O(n \log n)$-time upper bound for simple polygons. We observe that it is possible to construct the bounded Voronoi diagram of a simple polygon P with n vertices and r reflex angles within P in time $O(n \log r)$ provided a triangulation of P is given. Next, we introduce the concept of approximation of the bounded Voronoi diagram of a simple polygon and derive upper time-bounds on the construction of the diagram in terms of a parameter characterizing the goodness of the approximation and the number of vertices. The latter approach has an analogy in recent exhaustive research on presortedness. It yields fast algorithms for the bounded Voronoi diagrams of polygons for which good approximations can be build easily.

2. Bounded Voronoi diagrams for polygons with few reflex angles

A *partition* of a simple polygon P is a finite set C of convex polygons lying within P such that the insides of the polygons are pairwise disjoint and their union covers P. C is said to be convex if each polygon in C is convex.
Given a triangulation of P, we can cancel each of the reflex angles in P by drawing at most two appropriate edges from the triangulation incident to the vertex of the angle. Hence, we have:

Lemma 2.1: Given a triangulation of a simple polygon P with n vertices and r reflex angles, we can construct a convex partition of P consisting of $O(r)$ convex polygons in time $O(n)$.

Combining the above lemma with the linear-time algorithm for the Voronoi diagram of a convex polygon from [AGSS], and Chazelle's polygon cutting theorem [C,L85], we obtain our first result.

Theorem 2.1: Let P be a simple polygon with n vertices and r reflex angles. Given a triangulation of P, we can compute $Vorb(P)$ within P in time $O(n \log r)$.
Proof: Assign weight 1 to each vertex of a reflex angle in P and weight 0 to the remaining vertices of P. Recall Chazelle's theorem on polygon cutting in its general weighted form [C,L85]. By [C,L85], given a triangulation of P, we can find a diagonal that splits P into two sub-polygons, each of total weight not greater than two thirds of the total weight of P plus one, and also the whole family of such diagonals for sub-polygons recursively created. everything in linear time.
The diagonal together with the diagonal family can be structured into a binary tree, say U. where a diagonal d' is a child of a diagonal d if d' splits one of the sub-polygons resulting from drawing d. Let h be the height of U. Observe that $h = O(\log r)$.
Let A be the partition of P into sub-polygons induced by diagonals from U. Each sub-polygon a in A has $O(1)$ reflex angles. We can trivially partition each of the sub-polygons a into $O(1)$ convex polygons by Lemma 2.1, correspondingly extending U by $O(1)$ levels. Let B be the resulting convex partition of P.
We compute $Vorb(P)$ within P in a bottom-up manner. First, we apply the linear-time algorithm given in [AGSS] to compute the diagram for each of the convex polygons corresponding to the leaves in the extended U. Next, we move upwards U, computing at each non-leaf node of T the bounded Voronoi diagrams of the sub-polygons corresponding to the sons. The merging can be done in time proportional to the sizes of the diagrams, and consequently proportionally to the sizes of the sub-polygons, see [LL86]. Actually, in [LL86], Lee

and Lin showed that the merging of the straight-line duals to the bounded Voronoi diagrams, the so called generalized Delaunay triangulations, can be done in the polygon case in linear time. As there are $O(\log r)$ levels in U, and the total size of the sub-polygons corresponding to a given level is $O(n)$, the whole procedure takes $O(n \log r)$ time.Q.E.D.

3. Constructing the bounded Voronoi diagram from its approximation

For a non-negative integer m, a partition of a simple polygon P is called an m-approximation of $Vorb(P)$ if for each vertex of v the region $P(v)$ of v within P intersects at most m edges in the partition.
Suppose a convex m-approximation of $Vorb(P)$ is given. Our main tool in designing a fast algorithm for $Vorb(P)$ is the possibility of computing the intersection of the diagram with a convex sub-polygon b efficiently provided that the sites whose regions in $Vorb(P)$ cover b are known. Note here that *the above intersection is equal to the intersection of the standard Voronoi diagram of the sites with b*. The possibility of the efficient computing of the intersection of the diagram with b is derived from a generalization of the linear-time algorithm for the Voronoi diagram of a convex polygon discussed on p. 45 in [AGSS].

Lemma 3.1: Let b be a convex polygon in the plane, and let S be a set of points $p_1,...,p_k$ in the plane such that for $i = 1,...,k-1$, a continous piece of the boundary between the regions of p_i, p_{i+1} in the Voronoi diagram of S crosses the boundary of b clockwise around b. The intersection of b with the Voronoi diagram of S can be constructed in linear time.
Proof: It follows from our assumptions about $p_1,...,p_k$ and b that for $i = 1,...,k-1$, p_i and p_{i+1} are Delaunay neighbours, i.e. they are adjacent in the Delaunay triangulation of S. Apply the standard lifting map $\mu(x,y) \rightarrow (x,y,x^2+y^2)$ to S. It follows from the known relationship between the Delaunay triangulation of S the convex hull of $\mu(S)$ that for $i = 1,...,k$, $(\mu(p_i),\mu(p_{i+1}))$ is an edge of the convex hull (see [E]). Hence, the points p_i, $i = 1,...,k$, satisfy the following condition:
(1) For $i = 1,...,k-1$, there exists a plane which includes (p_i,p_{i+1}) and leaves all points in S different from p_i and p_{i+1} in the same halfplane.
Consider any subsequence $p_{i_1},...,p_{i_l}$ of the sequence $p_1,...,p_k$. Let S' be the set of elements in the subsequence. It follows from the convexity of b and the convexity of regions in the Voronoi diagram of S' that for $j = 1,...,l-1$ a continous piece of the boundary between the region of p_{i_j} and $p_{i_{j+1}}$ crosses the boundary of b. Hence, analogously as previously, we derive the following condition which is a generalization of condition (1).
(2) For $j = 1,...,l-1$, there exists a plane which includes $(p_{i_j},p_{i_{j+1}})$ and leaves all points in S' different from p_{i_j} and $p_{i_{j+1}}$ in the same halfplane.
By (2), we can compute the convex hull of $\mu(S)$ using a generalization of the algorithm for Voronoi diagrams of convex polygons given in [AGSS] (see p.45) in linear time. In turn, given the convex hull of $\mu(S)$, we can easily compute the Voronoi diagram of S in linear time by the above relationship [E]. The intersection of the Voronoi diagram of S with b can be found in linear time by the convexity of the regions in the Voronoi diagram and the convexity of b (see [PS]). Q.E.D.

Theorem 3.1 does not yield an obvious way of computing $Vorb(P)$ if a convex m-approximation of P is given. Simply the interactions between vertices of P on different sides on the edge between two convex parts can be two-way, i.e. regions in $Vorb(P)$ belonging to vertices on both sides of the edge can cross it. To start with, we shall consider a simpler situation where the interaction is only one-way. We can formalize such a situation as follows.
Let B be a partition of a simple polygon into convex parts. Consider the directed graph G whose nodes one-to-one correspond to the convex polygons in B, such that (v,v') is an edge in G iff the convex polygons corresponding to v, v', say b, b', share an edge which is crossed by regions of vertices lying on the side of b. We say that B is *acyclic* if G is acyclic.

Given an acyclic convex m-approximation of $Vorb(P)$, where m is small, we can efficiently compute $Vorb(P)$ according to the following theorem.

Theorem 3.1 Let P be a simple polygon on n vertices. Given an acyclic convex m-approximation of P consisting of $O(n)$ polygons, we can construct $Vorb(P)$ in time $O(nm)$.
Proof: Let B be the m-approximation of P. For each b in B, set $s(b)$ initially to the list of sites within b in clockwise order around b, i.e. the list of vertices of P within b in clockwise order around b. Set G to the directed acyclic graph corresponding to B. Next, iterate the following steps until G is empty.

Step 1. Pick a vertex v in G with indegree zero, delete it from G and set b to the member of B corresponding to v.
Step 2. Construct $Vor(s(b)) \cap b$.
Step 3. For each b' adjacent to b, appropriately augment $s(b')$ with the sites in $s(b)$ whose regions cross the edge between b and b' in $Vor(s(b))$, according to the clockwise order of their intersections with b.

Finally, glue all the computed fragments of Voronoi diagrams within b's in B and output the result as $Vorb(P)$.
We can prove by induction on the number of iterations that whenever $Vor(s(b)) \cap b$ is computed, the list $s(b)$ contains exactly all sites whose regions in $Vorb(P)$ intersect with b in clockwise order of their intersections with b. This combined with the convexity of b implies $Vor(s(b)) \cap b = Vorb(P) \cap b$. Thus, the above algorithm indeed produces $Vorb(P)$.
Let us estimate the time performance of the algorithm. The initialization preeceding the iterated sequence of instructions takes linear time. To implement Step 1, we maintain a list of vertices which have indegree zero currently. Whenever we delete a vertex v from G, we check whether the vertices in G it has been adjacent to already reached zero indegree. If so, we put them respectively on the list. It is easily seen that the total work done is Step 1 is proportional to the sum of degrees of vertices in the initial graph G which is $O(n)$ by the planarity of G. Consider Step 2. By Lemma 3.1, we can implement it in time $O(\#s(b))$. By the definition of m, a site can occur in at most m different lists $s(b)$. Therefore, Step 2 totally takes $O(nm)$ time. Step 3 takes time proportional to the number of edges of b and the number of edges in $Vorb(P) \cap b$. By the linear size of G and $Vorb(P)$, and the upper bound m on the depth of B, these numbers sum to $O(nm)$. Also, the final gluing of the produced pieces of $Vorb(P)$ can be done in time $O(\sum_{b \in B} \#s(b))$ which is $O(nm)$. Q.E.D.

In general, we cannot ensure acyclicity of a convex m-approximation of $Vor(P)$ avaible. We pay for it by slowing down our method of constructing $Vorb(P)$ as the following theorem shows.

Theorem 3.2: Let P be a simple polygon on n vertices. Given a convex m-approximation of P consisting of $O(n)$ polygons, we can construct $Vorb(P)$ in time $O(nm^2)$.
Sketch: Let B the convex m-approximation of P. The directed graph G corresponding to B may contain cycles. Therefore, comparing with the acyclic case, we have to spend more time to stabilize the lists $s(b)$ consisting of candidates for sites whose regions cover b in $Vorb(P)$.

For each b in B, we initially set $s(b)$ to the list of vertices of P within b in clockwise order around b as in the proof of Theorem 3.1. Next, we iterate the following instruction block until all the lists $s(b)$ stabilize.

1. For each b in B, compute $Vor(b(s)) \cap b$.
2. For each b in B, set $s(b)$ to the list of sites within b plus all the sites whose regions cross the boundary of b in $Vor(b'(s))$ (computed in Step 1), where $b' \in B$ shares an edge with b, in clockwise order of their regions intersections with b.

When all the lists $s(b)$ stabilize, we glue the Voronoi diagram fragments $Vor(s(b)) \cap b$, $b \in B$, and output as $Vorb(P)$.

For the sake of correctness proof for the above algorithm, we assume the following notation. For $b \in B$, $t(b)$ is the set of sites, i.e. vertices of P, whose regions in $Vorb(P)$ cover b. Next, we define the l-bounded Voronoi diagram of P, $Vorb_l(P)$, analogously as $Vorb(P)$ (see Introduction) additionally requiring that the segment (p, v) between a point p and a site v passes through at most l different convex polygons in B.

Now, by the induction on the number l of iterations of the block, we prove that the union of $Vor(s(b)) \cap b$, $b \in B$, forms $Vorb_l(P)$ just after the l-th iteration. It follows from the definition of m that after the m-th iteration, the lists $s(b)$, $b \in B$, never change. Hence, the algorithm terminates after performing $m + 1$ iterations of the block, and then gluing the diagram fragments.

Reasoning analogously as in the proof of Theorem 3.1, we infer that the initialization takes linear time, the l-iteration of the block takes time $O((size(Vorb_l(P)) + n)m)$, and gluing the pieces of $Vorb(P)$ $O(nm)$ time. Again, by induction on l, we can prove that $Vorb_l(P)$ has a linear number of edges. Thus, the algorithm runs in time $O(nm^2)$. Q.E.D.

4. Final Remarks

1. Algorithms for $Vorb(P)$ given in the proofs of Theorems 3.1 and 3.2 seems to admit efficient parallel versions.

2. A diagonal of a PSLG H is an open straight-line segment that neither intersects any edge of H nor includes any vertex of H and whose both endpoints are vertices of H. In [LL90], Levcopoulos and Lingas proved that given a PSLG H and $Vorb(H)$, we can find a shortest diagonal of H in linear time. Thus, Theorem 2.1 and 3.2 can be used to find a shortest interior diagonal of a simple polygon efficiently in case it has few reflex angles or a good m-approximation of $Vorb(P)$.

References

[A] A. Aggarwal, *Research topics in computational geometry*, Bulletin of EATCS 39 (October 1989), pp. 388-408.

[AGSS] A. Aggarwal, L. Guibas, J. Saxe, P. Shor, *A Linear Time Algorithm for Computing the Voronoi Diagram of a Convex Polygon*, in Proc. of the 19th ACM Symposium on Theory of Computing, New York, pp. 39-45,1987.

[C] B. Chazelle, *A Theorem on Polygon Cutting with Applications*. In Proc. of the 23rd IEEE FOCS Symposium.

[E] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Texts and Monographs in Computer Science, Springer Verlag, Berlin, 1987.

[LL86] D.T. Lee and A. Lin, *Generalized Delaunay Triangulation for Planar Graphs*, Discrete and Computational Geometry 1(1986), Springer Verlag, pp. 201-217.

[LL90] C. Levcopoulos, A. Lingas, *Fast Algorithms for Greedy Triangulation*. To appear in Proc. 2nd Scandinavian Workshop on Algorithm Theory, Bergen 1990, LNCS Springer Verlag.

[L85] A. Lingas, *On Partitionning Polygons*. In Proc. 1st SCM Symposium on Computational Geometry, Baltimore, 1985.

[L89] A. Lingas, *Voronoi Diagrams with Barriers and their Applications*, Information Processing Letters 32(1989), pp. 191-198.

[PS] F.P. Preparata and M.I. Shamos, *Computational Geometry, An Introduction*, Texts and Monographs in Computer Science, Springer Verlag, New York, 1985.

[TV] R.E. Tarjan and C.J. Van Wyk, *An $O(n \log\log n)$-time algorithm for triangulating a simple polygon*, SIAM Journal on Computing, 17 (1988), pp. 143-178.

[WS] C. Wang and L. Schubert, *An optimal algorithm for constructing the Delaunay triangulation of a set of line segments*, in: Proc of the 3rd ACM Symposium on Computational Geometry, Waterloo, pp. 223-232, 1987.