Finding Geodesic Voronoi Diagram of Points In the Presence of Rectilinear Barriers*

Yung H. Tsin
School of Computer Science
University of Windsor
Windsor, Ontario, Canada
N9B 3P4

Cao-An Wang
Department of Computer Science
Memorial University of Newfoundland
St.John's, Newfoundland, Canada
A1C 5S7

* Research supported by the Natural Sciences and Engineering Research Council of Canada under grants NSERC-A7811 and NSERC-A .

Extended Abstract

The *Voronoi diagram* has been proven to be a powerful tool since Shamos introduced it into computational geometry in the seventies. Variations from the *standard* Voronoi diagram have then been introduced and investigated. One of these variations is the Voronoi diagram of a set of points (called sites) constrained by rectilinear barriers. In this direction, the shortest distance between two points is measured by their *geodesic* which is the shortest path between them that does not cross any barrier. The corresponding Voronoi diagram is called a *geodesic Voronoi diagram* which is a planar subdivision in which each region is associated with a site and the points in the interior of each region has a shorter geodesic to the associated site than to any other sites in S. Aronov[1] consider the case where the rectilinear barriers form the boundary of a simple polygon enclosing the points. He presented an $O((n+m)\log^2(n+m))$ time and $(n+m)$ space algorithm where n is the number of points and m is the number of barriers. In this paper, we consider the case where the rectilinear barriers is a set of parallel line segments. We present an algorithm which takes $O((n+m)\log(n+m))$ time and $O(m+n)$ space. The algorithm is based on the plane-sweep paradigm and is worst-case optimal in both time and space. Both Aronov's and our results are generalization of Lee and Preparata's earlier work on Euclidean Shortest Paths[2]. In addition, our result also generalizes that of Fortune[3]. Our method is a combination of *plane-sweep* and *divide-and-conquer*. The algorithm consists of two plane-sweeps. The first (resp. second) sweep advances from left to right (resp. right to left) along the x-axis to construct a left-to-right (resp. right-to-left) Shortest Path Map. The divide-and-conquer procedure is a simple top-down non-recursive procedure which completes the geodesic Voronoi diagram based on the two Shortest Path Maps.

1. Definitions and Preliminary Results

We shall denote the sites by a set of n points $S = \{ s_1, s_2, \ldots, s_n \}$ and the barriers by a set of m parallel line segments $L = \{ l_1, l_2, \ldots, l_m \}$. Without loss of generality, we shall assume that the barriers are mutually disjoint and are perpendicular to the x-axis and that no site lies in the interior of a barrier. For each $l_j \in L$, $1 \le j \le m$, we let $l_j = \overline{p_{2j-1}p_{2j}}$ where p_{2j-1} and p_{2j} are the upper and lower end-points of l_j, respectively. For each $s_i \in S$, let $x(s)$ ($y(s)$ respectively) be the x-coordinate (y-coordinate, respectively) of s and for each $l_j \in L$, let $x(l_j) = x(p_{2j-1})$. and $y(l_j) = y(p_{2j-1})$. The binary relation $\lessdot$ is a partial order defined in $S \cup L$ such that $\beta_i \lessdot \beta_j$ iff $x(\beta_i) < x(\beta_j)$ or $x(\beta_i) = x(\beta_j)$ and $y(\beta_i) < y(\beta_j)$.

Definition: Let $d_g(p,q)$ denote the geodesic distance between two points p and q. The **geodesic Voronoi diagram** of a set of sites S in the presence of a set of line segments L, denoted by $Vor(S,L)$, is a collection of **Voronoi cells** $\{ V(s_i) \}_{s_i \in S}$ such that $V(s_i) = \{x \mid d_g(x, s_i) \le d_g(x, s_j), \forall s_j \in S, i \ne j \}$. We usually refer to the union of the closure of the Voronoi cells as Voronoi diagram. A **Voronoi vertex** is a point on the closure which is equidistant to at least three sites. A **Voronoi edge** is a piece of the closure delimited by a pair of Voronoi vertices and containing no other Voronoi vertex in its interior. [**Remark:** As with [2], we assume without loss of generality that the sites are in *general position* so that the closure of $V(S,L)$ does not contain any endpoint of L.]

Let α_i, α_j be two points and $w(\alpha_i)$, $w(\alpha_j)$ be two real numbers associated with α_i and α_j. The **bisector of α_i and α_j**, denoted by $B(\alpha_i,\alpha_j)$, is the locus of the points p such that $d(p,\alpha_i) + w(\alpha_i) = d(p,\alpha_j) + w(\alpha_j)$. Note that when $w(\alpha_i) = w(\alpha_j) = 0$, the above definition of bisector is the same as the usual one.

Lemma 1[2]: *Let $v_0\ v_1\ v_2 \ldots\ v_k$ be a geodesic between* p *and* q. *Then $v_t \in \{ p_j \mid 1 \le j \le 2m \} \cup \{ p,q \}$, $0 \le t \le k$, where the p_j's are the end-points of the barriers.*

Lemma 2[2]: *The geodesic between any two points is monotone w.r.t. the x-axis.*

We generalize the notion of SPM (Shortest Path Map) of Lee and Preparata[2] as follows:

Definition: Let $S^* = \{ \alpha_j \mid 1\le j\le n+2m \} = S \cup \{ p_j \mid 1\le j\le 2m \}$. For each point $\alpha_j \in S^*$, a label $s(\alpha_j)$, called the **source of** α_j, and a real number $w(\alpha_j)$, called the **weight of** α_j are associated with α_j. $s(\alpha_j)$ is the site that is the closest to α_j with $x(s(\alpha_j)) \le x(\alpha_j)$. $w(\alpha_j)$ is the distance between α_j and $s(\alpha_j)$. The **Left-to-Right Shortest Path Map for S*, LR-SPM(S*)**, is a partition of the closed right half-plane, defined by L_1 (the vertical line passing throught s_1), into regions $R(\alpha_j)$, $1\le j\le n+2m$, such that region $R(\alpha_j)$ is associated with the point $\alpha_j \in S^*$ and is the locus of the points p such that α_j is visible from p and $x(\alpha_j) \le x(p)$ and $d(p,\alpha_j) + w(\alpha_j) \le d(p,\alpha_k) + w(\alpha_k)$ for all α_k which are visible from p and satisfy $x(\alpha_k) \le x(p)$. The **Right-to-Left Shortest Path Map (RL-SPM)** is defined similarly.

In the above definition, we have assumed without loss of generality that s_1 is the left-most element in S^*. From Lemmas 1 and 2, it is easily verified that the regions in LR-SPM(S^*) are separated by the bisectors between points in S^*, the barriers from L and sections of the vertical lines passing throught the sites. Each of the bisectors is a section of a hyperbola or a straight line segment (a degenerated hyperbola). To be more specific, (*a*) for each α_i which is a site, say s_i, if v_{hi} (v_{ik}, resp.) is a point above (below, resp.) s_i in region $R(\alpha_h)$ ($R(\alpha_k)$, resp.) such that $w(\alpha_h) + d(\alpha_h, v_{hi}) = d(s_i, v_{hi})$ ($w(\alpha_k) + d(\alpha_k, v_{ik}) = d(s_i, v_{ik})$, resp.), then the initial portion of $R(s_i)$ is bounded above by bisector $B(\alpha_h, s_i)$, bounded below by $B(s_i, \alpha_k)$ and bounded on the left by the section of the vertical line passing through s_i intercepted by v_{hi} and v_{ik}. (Note: v_{hi} or v_{ik} may not exist.) (*b*) For each α_i which is an end-point of a barrier, say l_i, if $\alpha_i = p_{2i-1}$ (p_{2i} resp.) falls into the region $R(\alpha_h)$ ($R(\alpha_k)$, resp.), then the initial portion of $R(p_{2i-1})$ ($R(p_{2i})$, resp.) is bounded above by $B(\alpha_h, p_{2i-1})$ ($B(p_{2i-1}, p_{2i})$, resp.); bounded below by $B(p_{2i-1}, p_{2i})$ ($B(p_{2i}, \alpha_k)$, resp.) and bounded on the left by the section of l_i intercepted by the two bisectors. For each point p at which two or more bisectors meet, the following lemma ensures that the bisectors are continued by exactly one *new* bisector in the right-hand side of the vertical line passing throught p.

Lemma 3: *Let p be a point at which two or more bisectors meet in a LR-SPM (RL-SPM respectively). There is exactly one bisector originated from p and extends into the right-hand (left-hand, respectively) side of the vertical line passing throught p. This new bisector depends only on the two extreme intersecting bisectors.*
Proof: Similar to the proof of Lemma 5 in [2]. □

Lemma 4: *Every bisector is montone w.r.t. the x-axis in LR-SMP(S*) (RL-SPM(S*) resp.).*
Proof: Similar to the proof of Lemma 4 in [2]. □

Lemma 5: *Let s_i be a site and L_i be the vertical line passing through s_i. Let $B(\alpha_h, \alpha_k)$ be a bisector which intersects L_i at v_{hk} in the LR-SPM(S*). Then $d(s_i, v_{hk}) \le w(\alpha_h) + d(\alpha_h, v_{hk})$ iff the line segment $\overline{s_i v_{hk}}$ lies completely within the region of s_i in the LR-SPM(S*).*
Proof: Omitted due to space. □

2. Constructing the LR-SPM an RL-SPM

Since the construction of the two shortest path maps are similar, we shall consider only the construction of LR-SPM. Our construction of the LR-SPM generalizes Lee and Preparata's algorithm for the *single source shortest path with barriers problem*[2]. As with their algorithm, we uses the plane-sweep method and two data structures Q and T.

Q is a *priority queue* in which the elements (points) are arranged according to the total order $\lessdot$. An element in Q can be (i) a site of S or (ii) a barrier from L or (iii) an **event-point** which is the intersection of two bisectors. In case (ii), the coordinates of the two end-points of the barrier are stored along with the barrier in Q and in case (iii), the two bisectors are stored along with the event-point. A **Virtual event-point** is an event-point of which one of the two bisectors is deleted. Initially, Q contains the sorted set $S \cup L - \{s_1\}$.

T is a *balanced binary searched tree* which is used to record the status of the sweep line. It contains the set of bisectors which intersect the sweep line. The bisectors are ordered in T according to their vertical ordering along the sweep line from bottom to top. For each bisector B, a variable $X(B)$ is associated with it which defines the x-coordinate of its right end. We say that the intersection, v, of two bisectors B and B' is **valid** if it is to the right of the sweep line and $x(v) \leq \min(X(B), X(B'))$. Initially, T is empty.

Algorithm: Construct LR-SPM(S*);

while (*Q is not empty*) **do**

pop p out of Q. (* The sweep line is position at p *)

Case (I): p is a barrier $l_j = \overline{p_{2j-1}p_{2j}}$.

1. Determine the regions $R(\alpha_h)$ and $R(\alpha_k)$ that contain p_{2j-1} and p_{2j}, respectively by searching T.
   Set $s(p_{2j-1}) \leftarrow s(\alpha_h)$; $s(p_{2j}) \leftarrow s(\alpha_k)$;
   Set $w(p_{2j-1}) \leftarrow w(\alpha_h) + d(\alpha_h, p_{2j-1})$; $w(p_{2j}) \leftarrow w(\alpha_k) + d(\alpha_k, p_{2j})$;
2. Compute (the equations of) bisectors $B(\alpha_h, p_{2j-1})$, $B(p_{2j-1}, p_{2j})$, $B(p_{2j}, \alpha_k)$ and insert them into T.
   Set $X(\,B(\alpha_h, p_{2j-1})\,) \leftarrow X(\,B(p_{2j-1}, p_{2j})\,) \leftarrow X(\,B(p_{2j}, \alpha_k)\,) \leftarrow +\infty$.
   Delete all the bisectors that are truncated by l_j from T.
3. Let B_5 (resp. B_1) be the bisector directly above $B(\alpha_h, p_{2j-1})$ (resp. directly below $B(p_{2j}, \alpha_k)$) and $B_2 = B(\alpha_h, p_{2j-1})$, $B_3 = B(p_{2j-1}, p_{2j})$, $B_4 = B(p_{2j}, \alpha_k)$. (Note: B_1, B_5 may not exist).
   Compute $P = \{q_l \mid q_l$ is the *valid* intersection of B_l and B_{l+1} with smallest x-coordinate, $1 \leq l \leq 4\}$.
   for l := 1 to 4 **do if** ($x(q_l)$ exists and $x(q_l) \leq \min\{\, x(q_{l-1}), x(q_{l+1}) \,\}$)
   then insert x_l into Q; (*at most two q_l's are inserted*)

Case (II): p is a site s_i.

1. Set $w(s_i) \leftarrow 0$; $s(s_i) \leftarrow s_i$;
2. Determine the region $R(\alpha_h)$ containing a point u above s_i so that $d(s_i, u) = w(\alpha_h) + d(\alpha_h, u)$
   if ($R(\alpha_h)$ exist) **then** compute the bisector $B(\alpha_h, s_i)$ and insert it into T;
   set $X(\,B(\alpha_h, s_i)\,) \leftarrow +\infty$;
3. Determine the region $R(\alpha_k)$ which contains a point v below s_i so that $d(s_i, v) = w(\alpha_k) + d(\alpha_k, v)$
   if ($R(\alpha_k)$ exist) **then** compute the bisector $B(\alpha_k, s_i)$ and insert it into T;
   set $X(\,B(\alpha_k, s_i)\,) \leftarrow +\infty$;
4. Update Q as in step 3 of **case** (I) using the bisectors computed in Steps 2 and 3.

Case (III): p is an event-point which is the intersection of $B(\alpha_h, \alpha_j)$ and $B(\alpha_j, \alpha_k)$.

1. **case** // both $B(\alpha_h, \alpha_j)$ and $B(\alpha_j, \alpha_k)$ are deleted //: discard p;
   determine the intersection of the bisector directly above $B(\alpha_h, \alpha_j)$
   and the bisector directly below $B(\alpha_j, \alpha_k)$;
   if the intersection exists and is valid then insert it into Q; proceed to **endwhile**;
   // only one of $B(\alpha_h, \alpha_j)$ and $B(\alpha_j, \alpha_k)$ is deleted //:
   discard p and extend the surviving bisector (i.e. Set its X variable to $+\infty$)
   let B be the surviving bisector;
   // none of $B(\alpha_h, \alpha_j)$ and $B(\alpha_j, \alpha_k)$ is deleted //: delete $B(\alpha_h, \alpha_j)$, $B(\alpha_j, \alpha_k)$ from T;
   compute $B(\alpha_h, \alpha_k)$ and insert it into T; set $X(\,B(\alpha_h, \alpha_k)\,) \leftarrow +\infty$;
   let B be $B(\alpha_h, \alpha_k)$;
   endcase;
2. Let q, q' be the intersections of B with the bisectors above and below B, respectively. If at least one of q, q' exists and *valid*, then insert the one with smaller x-coordinate into Q and

update the X variables of the two corresponding bisector accordingly.

endwhile

The correctness of the above constrcution can be easily verified by applying induction on with the use of Lemmas 4 and 5 and the structures of LR-SPM(S^*).

Lemma 11: *Constructing LR-SPM can be done in O((m+n)log(m+n)) time and O(m+n) space.*

Proof: It can be shown that the number of non-virtual event-points, the number of virture event-points and the number of bisectors processed in the course of constructing the LR-SPM(S^*) are all bounded by O(m+n). Since insertion and deletion can be done in O(log$|T|$) time for T and in O(log$|Q|$) time and for Q and there are O(m+n) event-points and bisectors, we immediately have $|T|$=O(m+n) and $|Q|$=O(m+n). Therefore, the insertion and deletion operations performed on T and Q can each be done in O(log(m+n)) time. For each site s_i, the time spent on handling s_i is O(log(m+n)) + $k_i$log(m+n), where k_i is the number of bisectors deleted for s_i. For each barrier l_i, the time spent on handling l_i is O(log(m+n)) + $k_i$log(m+n), where k_i is the number of bisectors deleted for l_i. For each event point p, the time spent on handling p is O(log(m+n)). Since there are O(m+n)) bisectors and event-points and each bisector can be deleted at most once, the total time required is thus O($n$log(m+n) + $\Sigma_{i=1}^{n} k_i$ log(m+n) + $m$log(m+n) + $\Sigma_{i=1}^{m} k_i$ log(m+n) + (m+n)log(m+n)) = O((m+n)log(m+n)). The space complexity is easily verified. □

3. Completing the Geodesic Voronoi Diagram

The LR-SPM(S^*) and RL-SPM(S^*) form a partial geodesic Voronoi diagram. Based on them, we can complete the construction of the geodesic Voronoi diagram. To do so, we use a *simple* divide-and-conquer strategy: We divide S into two subsets S_L and S_R of approximately equal sizes. Then with the help of LR-SPM(S^*) and RL-SPM(S^*), we determine the dividing curve which consists of all those edges of *Vor(S,L)* which separate the sites in S_L from those sites in S_R. We then apply the same procedure to S_L (S_R respectively) to construct those Voronoi edges which separate the sites in S_L (S_R respectively) into two half. The procedure is repeated until every set of sites is a singleton and the geodesic Voronoi diagram is completed.

To determine the dividing curve for S_L and S_R, we proceed as follows:

1. For each region $R(\alpha_j)$ in LR-SPM(S^*), triangulate $R(\alpha_j)$ in such a way that every triangle has α_j as a vertex. Triangulate the regions in RL-SPM(S^*) in a similar way.
2. Determine the bisector on the dividing curve which intersects the line segment joining the right-most site in S_L and the left-most site in S_R. Starting from that bisector, trace out the dividing curve using the triangulated LR-SPM(S^*) and the triangulated RL-SPM(S^*).

Unfortunately, the dividing curve may consists of several disjoint pieces. Therefore after two end-points of the dividing curve are research, i.e one piece of the dividing curve is determined, the sites on each side of that dividing curve have to be examined to ensure that no side contains a site from S_L and a site from S_R. If that happens, then the above procedure will be repeated to trace out another piece of the dividing curve. The process is repeated until all the sites from S_L are separated from those from S_R. Based on the fact that there are at most O(m+n) triangles in the triangulated LR-SPM(S^*) and RL-SPM(S^*), it can be shown that determining the dividing curve takes O(m+n) time. Since there are at most log$n$ iterations, the total time required to complete the geodesic Voronoi diagram is thus O((m+n)log(m+n)). The space complexity can be shown to be O(m+n). The details are omitted due to space limitation.