Time-Based Voronoi Diagram*

D.T. Lee, † Chung-Shou Liao, and Wei-Bung Wang ‡
Institute of Information Science
Academia Sinica, Nankang, Taipei 115, Taiwan.
e-mail:{dtlee,shou794@iis.sinica.edu.tw, b89105@csie.ntu.edu.tw}

Abstract

We consider a variation of Voronoi diagram, or time-based Voronoi diagram, for a set S of points in the presence of transportation lines or highways in the plane. A shortest time-distance path from a query point to any given point in S is a path that takes the least travelling time. The travelling speeds and hence travelling times of the subpaths along the highways and in the plane are different. M. Abellanas et al. [1] gave a simple algorithm that runs in $O(n \log n)$ time, for computing the time-based Voronoi diagram for a set of n points in the presence of one highway in the plane. We consider a generalization of this problem to the case when there are two or more highways. We give a characterization of this problem and present an $O(n \log n)$ time algorithm for the problem where there are two highways. The algorithm can be easily extended to multiple highways if a certain intersection condition of highways holds.

1 Introduction

The Voronoi diagram is a very versatile and well-studied geometric construct in computational geometry[2, 3, 4, 5, 6, 7, 8]. A traditional underlying distance measurement of Voronoi diagram is the Euclidean distance. Given a set S of disjoint line segments, each of which may degenerate into a point, the Voronoi diagram for S in the plane is a partition of the plane into Voronoi regions, each of which corresponds to an element of S and is the locus of points closest (in terms of Euclidean distance) to the element than to any other element of S. It is known that the Voronoi diagram for S can be computed in $O(n \log n)$ time[8], where n is the size of S. In 2003, M. Abellanas et al. [1] introduced the traveling time-distance model by considering a set of points in the presence of a transportation line, or highway. It is assumed that the traveling speed

*Supported in part by the National Science Council under the Grants NSC-93-2213-E-001-013, NSC-92-3112-B-001-018-Y, NSC-92-3112-B-001-021-Y, NSC-93-2422-H-001-0001, and NSC 93-2752-E-002-005-PAE

†Also with the Dept. of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan

‡Dept. of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan

v_0 in the plane is different or slower than that on the highway. Instead of Euclidean distance, travelling time, i.e.. distance over speed, is used in the Voronoi diagram for one highway model. They presented an $O(n \log n)$ time algorithm for computing the Voronoi diagram of a set of n points in the plane in the presence of one highway. We extend the result by considering a multiple highway model as follows.

- All highways are straight lines, $L_1, L_2, \ldots, L_k, k > 0$.
- Travellers can enter the highways at any point and travel in both directions. The travelling speed allowed on highway L_i is v_i for all i.
- Off the highways travellers can move freely in the plane, and the travelling speed in any directions is $v_0 \ll v_i, i = 1, 2, \ldots, k$.

We show that the Voronoi diagram for two-highway model of a set of n points can also be computed in $O(n \log n)$ time. We further give some condition, under which generalization of our result for two-highway model to multiple-highway model can be done easily. In the next section we review the one-highway model result and transform the new time-distance model to the ordinary Euclidean distance, so the previously known results for traditional Voronoi diagrams can be used. In Section 3, we provide an $O(n \log n)$ algorithm for the two-highway model in which the two highways satisfy a certain angle condition. We extend this method and present in Section 4 an $O(k^3 \log k + k^2 n \log n)$ time algorithm for the multiple-highway model, provided that the highways pairwise satisfy the good angle condition. In Section 5, the two-highway model in general is also solved in $O(n \log n)$ time by case analysis. We point out the difficulty of the problem for the multiple-highway model and leave this problem for future study.

2 Preliminaries

Consider a set S of n points, $p_1, p_2, \ldots, p_n$, called *sites*, in the plane H. The time-distance $d_t(q, p_i)$ between any point q and p_i is defined as $d_t(q, p_i) = d(q, p_i)/v_0$, where $d(q, p_i)$ denotes the Euclidean distance between q and p_i and v_0 is the travelling speed in H. The locus of points closest to p_i in time-distance among all sites in S is the Voronoi region of p_i, denoted $Vor_t(p_i, S)$, or $Vor_t(p_i)$ when S is assumed. That is, $Vor_t(p_i, S) = \{q | d_t(q, p_i) \leq d_t(q, p_j), j \neq i, p_i, p_j \in S\}$. The collection of Voronoi regions for all sites in S is called the time-based Voronoi diagram, of S, denoted $Vor_t(S)$. If v_0 is the same everywhere in the plane, the time-based Voronoi diagram

$Vor_t(S)$ is the same as the ordinary Voronoi diagram of S, denoted by $Vor(S)$.

In the plane we assume there exist k lines, $L_1, L_2, \ldots, L_k$, $k > 0$, as highways, where each of which L_i is associated with a travelling speed $v_i \gg v_0$, for $i = 1, 2, \ldots, k$. Let us first review the $O(n \log n)$ time algorithm given by M. Abellanas et al. [1] for computing the time-based Voronoi diagram for the case when the number of highways is one.

Without loss of generality, we assume the highway L coincides with the x-axis, and its speed allowed is $v \gg v_0$. The time-distance between p and q in the presence of L can be transformed into Euclidean distance as follows. Assume p is above L. Suppose q is above L, and q^L denotes the reflection of q with respect to L. Draw two half-lines, denoted $+\hat{q}^L$ and $-\hat{q}^L$, emanating from q^L of slopes $+\tan\alpha_L$ and $-\tan\alpha_L$ respectively, where $\sin\alpha_L = \frac{v_0}{v}$. $+\hat{q}^L$ is called the plus-hat of q^L, and $-\hat{q}^L$ is called the minus-hat of q^L. Figure 1 shows the case when q lies on L, in which case $q^L = q$. The time-distance between p and q is defined to be the minimum of $d(p,q), d(p,q_\ell)$ and $d(p,q_r)$, where $q_\ell \in +\hat{q}^L$ and $d(p,q_\ell)$ is $\min_{x\in+\hat{q}^L}\{d(q,x)\}$, and $q_r \in -\hat{q}^L$, and $d(p,q_r)$ is $\min_{x\in-\hat{q}^L}\{d(q,x)\}$.

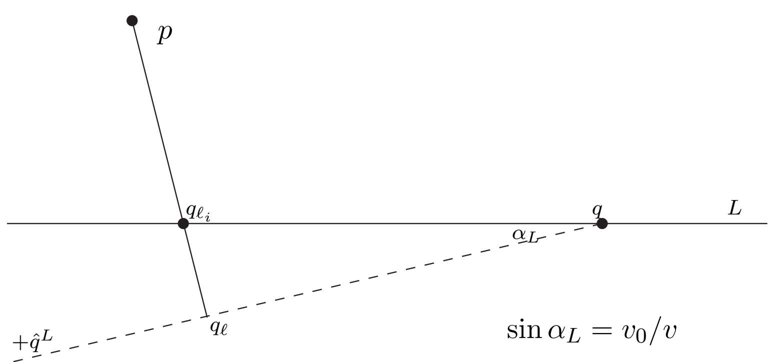

Figure 1: Enter the highway at q_{ℓ_i}.

Figure 2 shows the time taken by the path from p to q_r equals the time taken by the three-segment path from p to q_{r_i}, along the highway from q_{r_i} to q_{r_o} and from q_{r_o} to q. The three-segment path is simply referred to as a *highway* path.

In other words, point q is split into three different objects, q, $+\hat{q}^L$ and $-\hat{q}^L$. We can define the locus of points p that are equidistant to q and to $+\hat{q}^L$, i.e., $d(p,q) = d(p,+\hat{q}^L)$. This is the bisector of q and $+\hat{q}^L$ and it is a parabola, with q as the focus and $+\hat{q}^L$ as the directrix. Similarly we define another bisector, which is also a parabola, defined by q and $-\hat{q}^L$.

In case when q lies on the opposite side of L as p, then the time-distance between p and q is

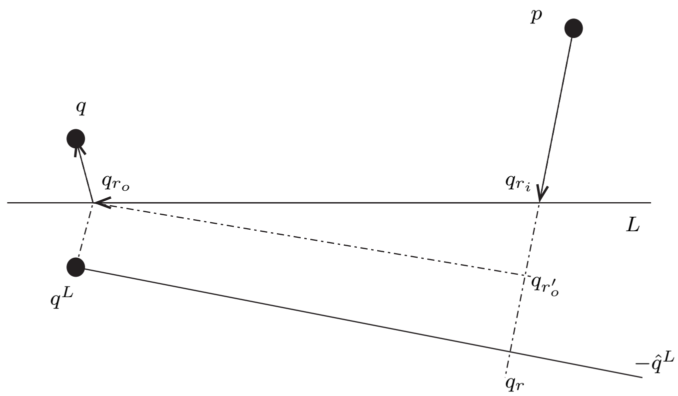

Figure 2: The time-distance of three-segment path equals the time-distance from p to $-\hat{q}^L$.

the smaller of $d(p, q_\ell)$ and $d(p, q_r)$, where $q_\ell \in +\hat{q}$ and $d(p, q_\ell)$ is $\min_{x \in +\hat{q}}\{d(q, x)\}$, and $q_r \in -\hat{q}$ and $d(p, q_r)$ is $\min_{x \in -\hat{q}}\{d(q, x)\}$. $+\hat{q}$ and $-\hat{q}$ are respectively the half-lines emanating from q of slopes $+\tan\alpha_L$ and $-\tan\alpha_L$. Note that the shortest time-distance path from p to q on the opposite side of L necessarily crosses the highway, and it is also a highway path. Similarly, point q is split into two objects, $+\hat{q}$ and $-\hat{q}$, and the region above L will be partitioned into three parts, each of which is associated with q, $+\hat{q}$ and $-\hat{q}$, respectively. For ease of reference we consider q, which is an endpoint of both $+\hat{q}$ and $-\hat{q}$, as an object, so that when q lies below L, we also split q into three objects.

The above transformation entails the following. The region above L is affected by three objects per site, i.e., the site q itself, plus the two half-lines associated with either q (when q is below L) or q^L (when q is above L).

The time-based Vonoroi diagram of a set S of sites in the presence of a highway L (positioned horizontally) is reduced to the following. The time-based Voronoi diagram in the half-plane above L will be the ordinary Voronoi diagram defined by the set of sites p that lie above L, their associated hats, $+\hat{p}^L$, $-\hat{p}^L$, and by the sets of sites q that lie below L and their associated hats $+\hat{q}$ and $-\hat{q}$. The time-based Voronoi diagram in the half-plane below L is defined similarly.

To sum up, let P^a and P^b denote the sets of objects used in defining the time-based Voronoi diagram above L and below L respectively. For convenience, the region above L is denoted as L^+, and the region below L is denoted as L^-.

Definition 2.1 $P^a = (\bigcup_{p \in L^+} (\{p\} \cup +\hat{p}^L \cup -\hat{p}^L)) \cup (\bigcup_{q \in L^-} (\{q\} \cup +\hat{q} \cup -\hat{q}))$.
$P^b = (\bigcup_{p \in L^+} (\{p\} \cup +\hat{p} \cup -\hat{p})) \cup (\bigcup_{q \in L^-} (\{q\} \cup +\hat{q}^L \cup -\hat{q}^L))$.

Theorem 2.2 *The Voronoi region for a site $p \in S$ with respect to S is given by*
For $p \in L^+$, $Vor_t(p, S) = L^+ \cap (Vor(p, P^a) \cup Vor(+\hat{p}^L, P^a) \cup Vor(-\hat{p}^L, P^a)) \cup L^- \cap (Vor(p, P^b) \cup Vor(+\hat{p}, P^b) \cup Vor(-\hat{p}, P^b))$.
For $p \in L^-$, $Vor_t(p, S) = L^- \cap (Vor(p, P^b) \cup Vor(+\hat{p}^L, P^b) \cup Vor(-\hat{p}^L, P^b)) \cup L^+ \cap (Vor(p, P^a) \cup Vor(+\hat{p}, P^a) \cup Vor(-\hat{p}, P^a))$.

Since the set P^a of objects below L consists of the hats associated with the sites above L and the hats associated with the sites below L (including the sites themselves), we can find the *envelope*[2] (as shown in Figure 3) defined by the collection of these hats. The plus-hats, and similarly the minus-hats, are all parallel lines. The sites or hats that are *below* or *dominated by* the envelope will not play any role in defining the time-based Voronoi diagram in the half-plane above L.

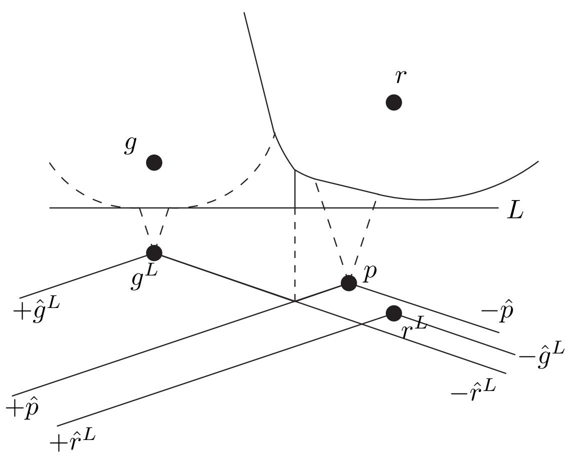

Figure 3: An illustration of the Voronoi diagram above L, and the envelope of the objects below L is shown in bold face.

3 Good Angle Condition for Two-Highway Model

Now we consider the two-highway model. Assume L_1 and L_2 intersect at origin O, and θ, where $0 < \theta \leq \pi/2$, is the angle between L_1 and L_2. p^{L_1} is the reflection of p with respect to L_1, and p^{L_2} is defined similarly. We recall $\sin \alpha_{L_1} = v_0/v_1$, $\sin \alpha_{L_2} = v_0/v_2$.

Lemma 3.1 *If $\alpha_{L_1} + \alpha_{L_2} = \theta$, the number of shortest time-distance paths between two points, $p \in L_1$ and $q \in L_2$ is infinite.*

Proof. (Sketch) Refer to Figure 4 (a) in which line h forms an angle α_{L_1} and α_{L_2} with L_1 and L_2, respectively. It is easily seen that $d_t(q,q\prime) = d_t(q,O) + d_t(O,q\prime)$ because $\sin\alpha_{L_1} = v_0/v_1$, $\sin\alpha_{L_2} = v_0/v_2$. □

Lemma 3.2 *If $\alpha_{L_1} + \alpha_{L_2} < \theta$, then the shortest time-distance path between two points, $p \in L_1$ and $q \in L_2$ must go through the origin O.*

Proof. (Sketch) Refer to Figure 4 (b) in which lines h_1 and h_2 form an angle α_{L_1} with L_1 and α_{L_2} with L_2, respectively. □

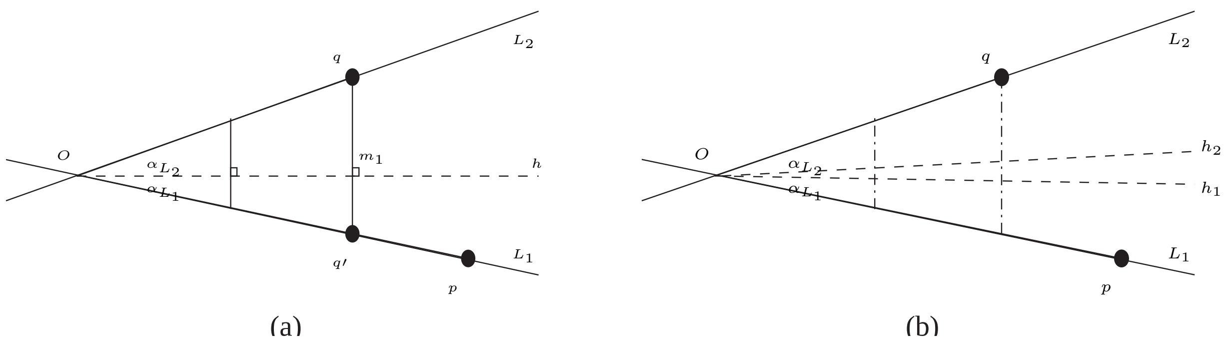

Figure 4: (a) The number of the shortest time-distance path between p and q is infinite. (b) The shortest time-distance path between p and q is unique and passes through O.

Lemma 3.3 *If $\alpha_{L_1} + \alpha_{L_2} > \theta$, the shortest time-distance path between two points, $p \in L_1$ and $q \in L_2$ will not go through the origin O.*

Proof. (Sketch) It is easily seen (Figure 5) that the path $p - s - t - q$ is better than $p - O - q$. In particular the shortest time-distance path is shown in bold-face line. □

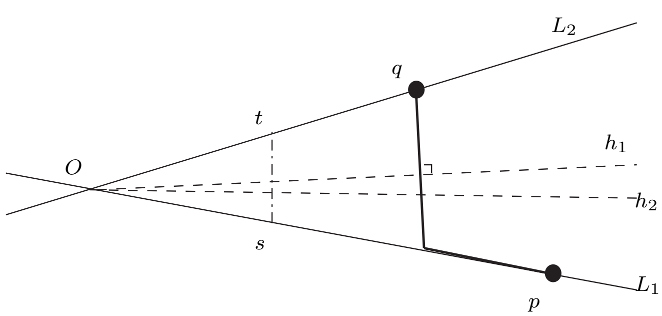

Figure 5: The unique shortest time-distance path between p and q does not go through O.

We shall assume in the following that the angle defined by L_1 and L_2 satisfies $\alpha_{L_1} + \alpha_{L_2} \leq \theta$. We shall refer to this intersection condition as **good angle condition**. When $\alpha_{L_1} + \alpha_{L_2} = \theta$ we assume that the shortest time-distance path between two points $p \in L_1$ and $q \in L_2$ always goes through O. In other words, if the shortest time-distance path between any two points passes through both highways, then it must pass through the intersection.

Without loss of generality we shall assume that highway L_1 coincides with the x-axis and the plane H is partitioned by L_1 and L_2 into four quadrants, and the i-th quadrant is denoted by $Q_i, i = 0, 1, 2, 3$.

Definition 3.4 *If p_O is the site closest to O, i.e., $d_t(O, p_O)$ is $\min_{p_j \in S} d_t(O, p_j)$, then p_O is called the O-**domination site**.*

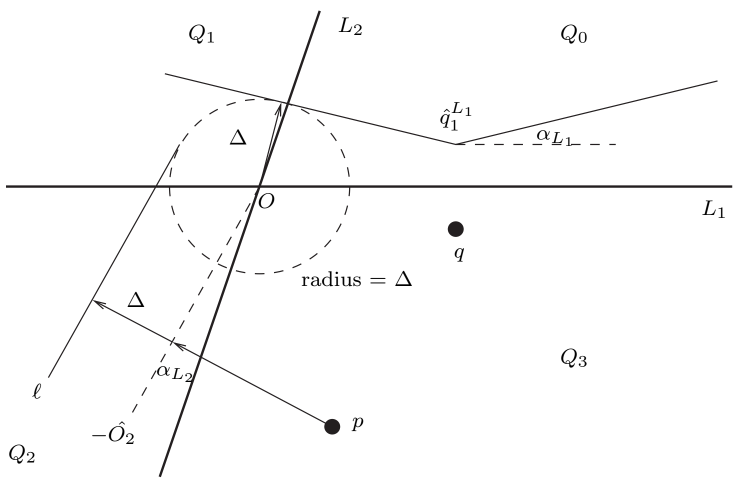

Figure 6: $d_t(p, q) = \min\{d(p, q), d(p, \ell)\}$

Lemma 3.5 *Suppose L_1 and L_2 satisfy the good angle condition. For any point $q \in Q_i$, if $d_t(q, p_i) = \min_{p \in S} d_t(q, p)$ and $p_i \in Q_{(i+2)mod4}$, then p_i must be the O-domination site.*

We now give a short description about the O-domination site, and its associated objects. As shown in Figure 6, suppose site q is the O-domination site and the shortest time-distance path from O to $q \in Q_3$ is via L_1, That is, $d_t(O, q) = \Delta = d(O, -\hat{q}_1^{L_1})$. Consider a query point $p \in Q_3$, and We want to find the shortest time-distance path from p to q. The time-distance from p to O is $d_t(p, O) = d(p, -\hat{O}_2)$. So the time-distance of a path from p to q via O is $d_t(p, O) + d_t(O, q)$. If we draw a line ℓ parallel to $-\hat{O}_2$ with a distance equal to Δ as shown in Figure 6, then $d_t(p, O) + d_t(O, q) = d(p, \ell)$. $d_t(p, q) = \min\{d(p, q), d(p, \ell)\}$. That is, line ℓ could be considered as an object derived from q in the same manner in which q splits into $\hat{q}_1^{L_1}$. In case the shortest time-distance path from O to q is via highway L_2, a similar line ℓ' parallel to $+\hat{O}_1$ with distance Δ may be derived. See Figure 7 for an illustration below.

In general, we have the following definition. Assume L_1 is positioned horizontally, and O is the intersection of L_1 and L_2. L_1 and L_2 partition the plane into 4 quadrants, $Q_i, i = 0, 1, 2, 3$. Let the line that borders quadrant Q_i and $Q_{(i+1)mod4}$ be denoted L_{i+} and the line that borders quadrant Q_i and $Q_{(i-1)mod4}$ be denoted L_{i-}. Note that L_{i+} is the same as $L_{(i+1)-}$. For instance, L_{0+} and L_{1-} denote the same line, which is line L_2, and L_{1+} and L_{2-} denote the same line, which is L_1. Let p_O be the O-domination site such that $d_t(O, p_O) = \Delta$, and $\Gamma_\Delta(O)$ denote the circle centered at O and of radius Δ.

Definition 3.6 *Let ℓ_1 be the line tangent to $\Gamma_\Delta(O)$ parallel to $+\hat{p}_{i+}^{L_{i+}}$ for any $p \in Q_i$, and ℓ_2 be another line tangent to $\Gamma_\Delta(O)$ parallel to $-\hat{q}_{i-}^{L_{i-}}$ for any $q \in Q_i$. The **Δ-distance-line-from-O** for p_O in Q_i is defined to be $\ell_i^\Delta(O) = \ell_1 \cup \ell_2$. It is simply called the **O-domination line** in Q_i.*

For any query point $p \in Q_i$, the objects into which the O-domination site P_O splits include the O-domination line in Q_i. Figure 7 illustrates the O-domination line in Q_3 with O-domination site p_O. The dotted part of the O-domination line wouldn't affect the Voronoi diagram, so will be omitted.

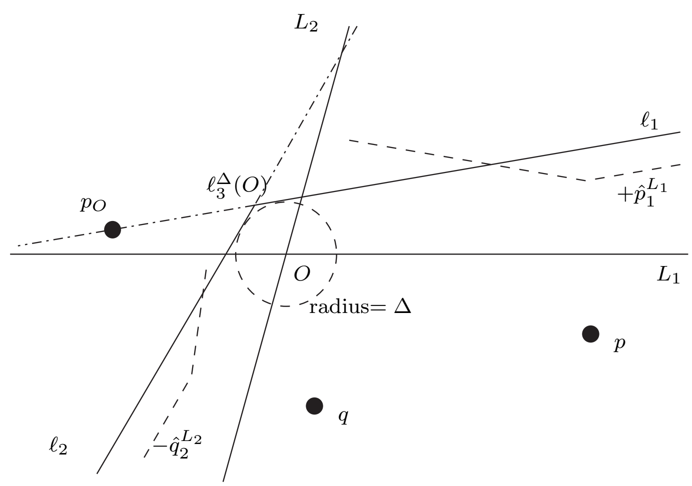

Figure 7: Δ-distance-line-from-O in Q_3

As we described earlier, any site $p \in Q_i$, except possibly the O-domination site, would not play any role in the time-based Voronoi diagram in $Q_{(i+2)mod4}$.

Following Theorem 2.2 we define the following *hats*. Let $\hat{p}_1$ denote the union of plus-hat $+\hat{p}_1$ and minus-hat $-\hat{p}_1$ with respect to L_1, when L_1 is assumed to be horizontally positioned. $\hat{p}_2$ is similarly defined with respect to L_2.

We summarize in **Algorithm2Line** (S, L_1, L_2) the algorithm for computing the Voronoi diagram for a set S of n sites in the presence of two highways, L_1 and L_2, when L_1 and L_2 satisfy the good angle condition.

Algorithm2Line (S, L_1, L_2)

Input: A set S of n sites, and two lines L_1 and L_2, which satisfy the good angle condition defined earlier.

Output: The time-based Voronoi diagram $Vor_t(S)$.

Method:

1. Find the O-domination site p_O and let $\Delta = d_t(O, p_O)$.
2. Compute the O-domination line in Q_i, $\ell_i^{\Delta}(O)$, for $i = 0, 1, 2, 3$
3. Compute the set P^i of objects used for constructing the Voronoi diagram in each quadrant Q_i for $i = 0, 1, 2, 3$. That is, $P^i = S \cup (\bigcup_{p \in Q_i} (\hat{p}_{i+}^{L_i+} \cup \hat{p}_{i-}^{L_i-}) \cup (\bigcup_{p \in Q_{(i+1) mod 4}} \hat{p}_{i+}) \cup (\bigcup_{p \in Q_{(i-1) mod 4}} \hat{p}_{i-}) \cup \ell_i^{\Delta}(O)$, for $i = 0, 1, 2, 3$.
4. Compute the (ordinary) Voronoi diagram in Q_i, i.e., $Vor(P^i) \cap Q_i$, for $i = 0, 1, 2, 3$.
5. Compute the time-based Voronoi region $Vor_t(p, S)$ for each site $p \in S$. That is, $Vor_t(p_O, S) = \bigcup_{i=0}^{3} (Vor(\ell_i^{\Delta}(O), P^i) \cup Vor(obj_{p_O}^i, P^i))$, and for $p \neq p_O$, $Vor_t(p, S) = \bigcup_{i=0}^{3} Vor(obj_p^i, P^i)$, where obj_p^i includes p itself and its associated hats in Q_i.

Theorem 3.7 *The Voronoi diagram for a set S of n sites in the presence of two highways L_1 and L_2 in the plane that satisfy the good angle condition, can be computed in $O(n \log n)$ time.*

Proof: The correctness of the algorithm follows from the above discussions and Lemma 3.5. In **Algorithm2Line** (S, L_1, L_2) the O-domination site can be computed in linear time. The time-based Voronoi diagram in quadrant Q_i is reduced to the problem of computing the (ordinary) Voronoi diagram for a set of sites $p \in Q_i$, and a set of line segments, obtained by the sets of hats associated with sites in Q_i, $Q_{(i+1) mod 4}$, and $Q_{(i-1) mod 4}$, and the O-domination line in Q_i

for the O-domination site. Since these sets of hats can be simplified by the notion of envelope, which can be obtained in $O(n \log n)$ time, as there are $O(n)$ hats in each quadrant, we can easily conclude that the Voronoi diagram in each quadrant can be computed in $O(n \log n)$ time, and that the total time complexity for computing the Voronoi diagram of n sites in the presence of two highways is $O(n \log n)$[8]. This completes the proof. □.

4 Good Angle Condition for Multiple-Highway Model

Now we can generalize this result to multiple highways. Assume that there exist k highways $L_i, 1 \leq i \leq k, k > 0$, and that they form an *arrangement*[3] of lines, partitioning the plane H into $O(k^2)$ cells.

The arrangement can be represented by a graph, $G = (V, E)$, where V denotes the set of intersections, and E denotes the set of edges connecting adjacent vertices along a line. Each edge, some of which is unbounded, borders two neighboring cells.

We shall as before, determine for each intersection (or vertex of G) the corresponding **intersection-domination site** similar to the O-domination site defined in the previous section. An intersection-domination site p_w of intersection $w \in V$ satisfies $d_t(w, p_w) = \min d_t(w, q) \forall q \in S$. p_w is called the w-domination site. We will also compute for each intersection w, the w-domination line associated with each cell.

Finally, for each cell we shall compute the set of objects associated with the sites that will define the time-based Voronoi diagram in the cell.

In Section 3 we know how to compute the O-domination site, its associated O-domination line in each cell (or quadrant) and the set of objects needed to define the Voronoi diagram in the cell. We shall use an iterative method by inserting the highways one at a time in order of non-descending speeds, and update the information needed to maintain the Voronoi diagrams. We shall index the set of highways $L_i, 1 \leq i \leq k, k > 0$ such that their associated speeds satisfy $v_1 \leq v_2 \leq \ldots, \leq v_k, k > 0$.

First of all, the graph $G_j = (V_j, E_j)$ that represents the line arrangements after the first $j < k$ lines are inserted can be maintained easily [7]. Now we briefly sketch how to determine the intersection-domination sites. For an intersection w, its domination site p_w could either lie in neighboring cells, or be propagated from neighboring intersection (u), i.e., p_u, when the shortest time path from p_w to w passes through u. In the latter case, $p_w = p_u$. Assume that in G_j, the intersection-domination sites have been obtained for each intersection in $V_j, j < i$. Suppose highway L_i is inserted and assume that the intersection w of L_i and L_w is on the edge $(u, z) \in E_j, u, z \in V_j$ (see Figure 8). w is incident with in general, four neighboring cells, C_0^w, C_1^w, C_2^w and C_3^w. Among all the sites in these four neighboring cells find a site p_w which is tentatively the w-domination site. Update p_w if $d_t(w, p_w) > d_t(w, p_u)$, or if $d_t(w, p_w) > d_t(w, p_z)$, where p_u and p_z are the u- and z-domination sites respectively. In this case, u (or z) is said to *dominate* w. This can be handled by a simple comparison of $d_t(w, p_w)$ and $d_t(u, p_u) + d_t(w, u)$ (or $d_t(z, p_z) + d_t(w, z)$). On the other hand, that is, if $d_t(w, p_w) < d_t(w, p_u)$ or $d_t(w, p_w) < d_t(w, p_z)$, the *domination* by w may propagate farther through u or z. Considering propagation from intersections on L_i to their neighbors, we build and maintain a min heap for propagation order.

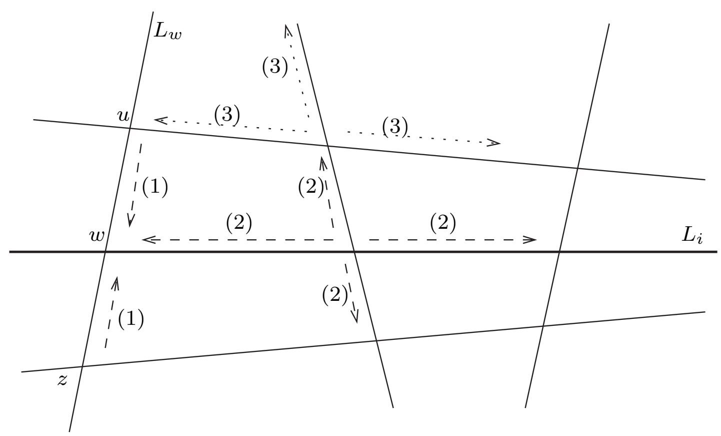

Figure 8: The propagation we consider is in this order.

Procedure *Intersection-Domination Sites Determination* /* The operation (1) in Figure 8. */

1. Sort all highways by their speed, and assume $v_1 \leq v_2 \leq \ldots \leq v_k$;

2. **for** $i = 1$ to k

   { Insert L_i into the plane H;

   for all intersections w on L_i

   { Find the nearest domination site p_w in neighboring cells;

   For the two neighboring intersections, u and z, not on L_i, if u (or z) *dominates* w, then $p_w \leftarrow p_u$ (or $p_w \leftarrow p_z$); }

   Call *Domination-Propagate*(G_i); }

In the next procedure, we define a set HOT, the elements of which are ordered pairs of vertices. Additionally, we define the cost function $c()$ as follows.

Definition 4.1 *For all u, let $c(u)$ denote the time-distance between intersection u and its current domination site p_u, and $c(u, v)$ the time-distance between u and v.*

Procedure *Domination-Propagate*(G_i) /* The operations (2), (3) in Figure 8. */

1. $HOT \leftarrow \{(u, v) : (u, v) \in E_i,\ u$ is on $L_i\}$;

2. **while** $HOT \neq \phi$

   { choose (u, v) such that $c(u) + c(u, v) = \min_{(j,k)\in HOT} \{c(j) + c(j, k)\}$;

   if $c(v) > c(u) + c(u, v)$

   { $c(v) \leftarrow c(u) + c(u, v)$ and $p_v \leftarrow p_u$;

   for all $w \neq u$ neighboring v

   { **if** $(w, v) \in HOT$

   $HOT \leftarrow HOT \setminus \{(w, v)\}$;

   else if $c(v) + c(v, w) < c(w)$

   $c(w) \leftarrow c(v) + c(v, w)$ and $HOT \leftarrow HOT \cup \{(v, w)\}$; }

   }

   $HOT \leftarrow HOT \setminus \{(u, v)\}$;

   }

We build a min heap for elements in HOT, so to push or pop the heap in $O(\log |HOT|)$ each time. It costs $O(i^2 \log i)$ time for procedure Domination-Propagate. The total time of intersection domination sites determination is $\sum_{i=1}^{k} O(i^2 \log i) = O(k^3 \log k)$. Finally, adding the time of computing the time-based Voronoi diagram of each cell, we have the total time complexity is $O(k^3 \log k + k^2 n \log n)$.

Theorem 4.2 *The time-based Voronoi diagram for a set of n sites in the presence of $k > 0$ highways, which pairwise satisfy the good angle condition, can be computed in $O(k^3 \log k + k^2 n \log n)$ time.*

5 General Condition for Two-Highway and Multiple-Highway Model

Now we briefly address the problem of two-highway model in which the two highways need not satisfy the good angle condition.

Lemma 5.1 *Let p, q be any two point on the plane. If the number of shortest time-distance path from p to q is finite, and the shortest time-distance path walks along both highways, then the path must walk through the intersection of two highways.*

Proof. (Omitted)

Lemma 3.5 does not hold if the good angle condition is not satisfied. For a site $q \in Q_i$, its time-based Voronoi region could intersect $Q_{(i+2)mod4}$. We thus need to consider all possible cases.

The following definition gives the set of objects that would be involved in the computation of time-based Voronoi diagram for quadrant $Q_i, i = 0, 1, 2, 3$. For quadrant Q_i, the sets of hats associated with sites in every quadrant may play a role in the computation. The notion of *envelope* and **Algorithm2Line** in Section 3 are still applicable, except that the size of the sets of objects has increased necessarily by a constant factor.

Definition 5.2 *When L_1 (positioned horizontally) and L_2 are in general position, the set P^i of objects involved in the computation of the time-based Voronoi diagram in cell Q_i is defined as follows.* $P^i = S \cup (\bigcup_{p \in Q_i} \hat{p}_{i+}^{L_{i+}} \cup \hat{p}_{i-}^{L_{i-}}) \cup (\bigcup_{p \in Q_{(i+1)mod4}} \hat{p}_{i+} \cup \hat{p}_{i-}^{L_{i-}}) \cup (\bigcup_{p \in Q_{(i+2)mod4}} \hat{p}_{i+} \cup \hat{p}_{i-}) \cup (\bigcup_{p \in Q_{(i+3)mod4}} \hat{p}_{i+}^{L_{i+}} \cup \hat{p}_{i-}) \cup \ell_i^{\Delta}(O)$.

There are two extreme cases in which the size of the sets of objects involved can be reduced further. They are: when two highways are parallel, and when the intersection angle θ is small and satisfies $\alpha_{L_2} > \alpha_{L_1} + \theta$.

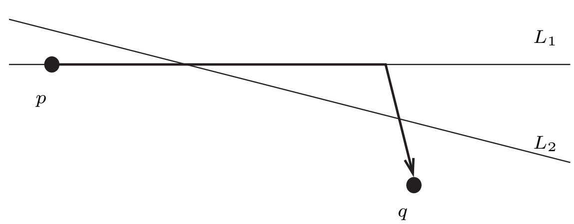

Figure 9: An example that shows L_1 nullifies L_2 when $\alpha_{L_2} > \alpha_{L_1} + \theta$

Figure 9 shows an example that if a shortest time-distance path walks along L_1, it will never walk along L_2. That is, L_1 nullifies L_2, when $\alpha_{L_2} > \alpha_{L_1} + \theta$. In this case the set of objects P^i for Q_i is the same as that in the general condition except $\ell_i^{\Delta}(O)$.

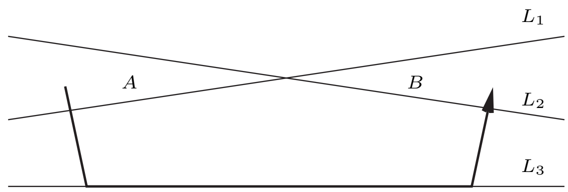

Figure 10: L_3 provides a new time-distance path from cell A to B.

Consider Figure 10. If there are only two highways L_1 and L_2, the shortest time-distance path from cell A to cell B must pass through the intersection of L_1 and L_2 (Lemma 5.1). However, adding L_3 will create a new kind of shortest time-distance paths without passing through any highway intersection. This makes it hard to determine or characterize what will be relevant in the construction of the time-based Voronoi region in a cell when multiple highways are present. Determining the intersection domination site for each intersection is yet another problem. We will leave this problem for future study.

Acknowledgement

We thank the anonymous reviewers for their comments on an earlier version of the paper.

References

[1] M. Abellanas, F. Hurtado, V. Sacristán, C. Icking, L. Ma, R. Klein, E. Langetepe, B. Palop : *Voronoi Diagram for Services Neighboring a Highway*, Information Processing Letters 86 (2003) 283-288.

[2] O. Aichholzer, F. Aurenhammer, D. Z. Chen, D. T. Lee and E. Papadopoulou, *Skew Voronoi Diagram*, Int'l J. Comput. Geometry and Applications, 9(3), June 1999, pp. 235-248.

[3] H. Edelsbrunner, *Algorithms in combinatorial Geometry*, Springer-Verlag New York, Inc., New York, NY, 1987.

[4] D. T. Lee, *Two Dimensional Voronoi Diagram in the L_p-metric*, J. ACM, Oct. 1980, pp. 604-618.

[5] D. T. Lee and R. L. Drysdale, *Generalization of Voronoi Diagram in the Plane*, SIAM J. Comput. Feb. 1981, pp. 73-87.

[6] E. Papadopoulou and D. T. Lee, *A New Approach for the Geodesic Voronoi Diagram of Points in a Simple Polygon and Other Restricted Polygonal Domains*, Algorithmica, 20(4), April 1998, pp. 319-352.

[7] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag New York, Inc., New York, NY, 1985.

[8] C. K. Yap, An $O(nlogn)$ algorithm for the Voronoi diagram of a set of simple curve segments. Discrete and Computational Geometry, Vol. 2, pp 365-393, 1987.