Voronoi Diagram for services neighboring a highway*

M. Abellanas
D. Matemática Aplicada
U. Politécnica de Madrid
Spain

F. Hurtado, V. Sacristán
D. Matemàtica Aplicada II
U. Politècnica de Catalunya
Spain

C. Icking, L. Ma
Prakt. Inform. VI
FernUni Hagen
Germany

R. Klein, E. Langetepe
Institut für Informatik I
U. Bonn
Germany

B. Palop
D. Informática
U. de Valladolid
Spain

14th May 2003

Abstract

We are given a transportation line where displacements happen at a bigger speed than in the rest of the plane. A shortest time path is a path between two points which takes less or equal time than any other. We consider the time to follow a shortest time path to be the *time distance* between the two points. In this paper, we give a simple algorithm for computing the Time Voronoi Diagram, that is, the Voronoi Diagram of a set of points using the time distance.

Keywords: Computational geometry, Algorithms, Voronoi diagrams, Transportation networks.

1 Introduction

Access time is obviously a most important issue for customers when choosing a service. When several suppliers provide similar services, a natural question arises: Which service can each customer reach faster? From the geometrical viewpoint modeling this situation requires the definition of a distance function taking into account the time for travelling between any two points in the plane, and the computation of the Voronoi Diagram for a set of points under this distance.

In order to define such a distance function, assumptions on the nature of the movements in the plane have to be made precise. Several variations on this subject are explored in [8]. The model we consider in this paper is the following:

- There is a big highway crossing some area which we will describe as a line in the plane.
- Travellers can enter the highway at any point and travel in it at speed v in both directions.
- Out of the highway travellers can move freely, and the travelling speed in any direction is $v_0 << v$.

*Partially supported by Proyecto MEC-DGES-SEUID-PB98-0933 and Accion integrada Alemania-España HA1999-0094. The first author is also supported by CAM-07T/0014/2001, the second and third authors by Gen. Cat. 2001SGR00224, and the fourth author by UAH2002/2001.

We call *Straight Line Transportation Model* the geometric framework obtained under the above set of assumptions.

We may assume without loss of generality that the transportation line L lies on the x-axis. Let us denote L^+ and L^- the halfplanes containing the points with non-negative and non-positive y-coordinates, respectively. This simplified model satisfies an obvious symmetry priciple, in the sense that any geometric consideration on points in L^+ has its dual (symmetric) for points in L^- and viceversa. Except otherwise stated, we deal with points in L^+ and leave implicit analogous considerations for points in L^-.

2 Time Distance

Given two points $p, q \in \mathbb{R}^2$, we say that a path γ between p and q is a *shortest time path*, $sp_t(p,q)$, if there is no other path between them taking strictly less time than the time required to follow γ. The *time distance* between p and q is the time required to follow any of the shortest time paths between them.

Given two points $p, q \in \mathbb{R}^2$, there is a simple way to decide which is the shortest time path between them. Let $v_0 = 1$ be the travelling speed out of L and let v be the speed in L. Note that, under this assumption, the Euclidean distance between the two points, $d(p,q)$, coincides with the time needed to follow the shortest path with endpoints p and q, if it does not intersect in any segment with L. Otherwise, Snell's law of refraction lets us find the best way of travelling.

Take $\alpha \in (0, \frac{\pi}{2})$ to be the angle such that $\sin\alpha = 1/v$. For a point $p \in L^+$, we denote by p_r (*right footpoint of* p) and p_l (*left footpoint of* p) the intersection of L with the lines through p whose slopes are $1/\tan\alpha$ and $-1/\tan\alpha$, respectively. We define the *network function*, $net : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ as follows. Let q be to the right of p. If $q \in L^+$ (resp. $q \in L^-$), let t be the intersection point of L with the segment qp_s (resp. pq), where p_s is the symmetric point of p with respect to L. Then,

$$net(p,q) = \begin{cases} d(p,p_r) + \frac{1}{v}d(p_r,q_l) + d(q_l,q) & \text{if } t \text{ is to the right of } p_r, \\ d(p,t) + d(t,q) & \text{otherwise.} \end{cases}$$

Given $p \in L^+$, let $S_l(p)$ (resp. $S_r(p)$) be the halfline with endpoint p_s and slope $\tan(\alpha)$ (resp. $-\tan(\alpha)$) to the left of p_s (resp. right). The following lemma, which can be proved using simple trigonometry, describes the network function between two points as the Euclidean distance between a point and a line.

Lemma 1 *Let $p \in L^+$ and let q be an arbitrary point in the plane such that t is to the right of p_r.*

$$net(p,q) = \begin{cases} d(q, S_r(p)), & \text{if } q \in L^+ \\ d(q, S_r(p_s)), & \text{if } q \in L^-. \end{cases}$$

Note that, if t is strictly between p_l and p_r, then the shortest path between p and q does not intersect L in a segment. Hence, we do not have to consider the value of $net(p,q)$, which is actually strictly greater than $d(p,q)$.

We characterize the *time distance*, $d_t : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ by the following lemma.

Lemma 2 *For any two points $p, q \in \mathbb{R}^2$,*

$$d_t(p,q) = \min\{d(p,q), net(p,q)\}.$$

From this characterization, it is easy to prove the following theorem:

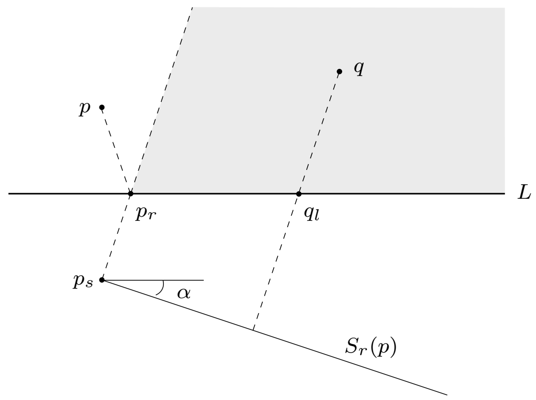

Figure 1: If q is in the shadowed region, then $net(p,q) = d(q, S_r(p))$.

Theorem 3 *The function* $d_t : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}_{\geq 0}$ *is a distance function.*

We end this section with a theorem which summarizes how can we compute the time distance between two points.

Theorem 4 *Let* $p \in L^+$ *and let* q *be an arbitrary point in the plane.*

$$d_t(p,q) = \begin{cases} \min\{d(q,p), d(q,S_r(p)), d(q,S_l(p)\} & \text{if } q \in L^+; \\ \min\{d(q,p), d(q,S_r(p_s)), d(q,S_l(p_s)\} & \text{if } q \in L^-. \end{cases}$$

3 Time Voronoi Diagram

In spite of the apparent simplicity of the Straight Line Transportation Model, most standard algorithms in computational geometry for computing basic structures cannot be easily adapted [1, 7], and matching optimal running times becomes surprisingly challenging. A main reason accounting for this fact is that the unit ball for the time metric has changing shape according to the position of its center, and it is not convex in the Euclidean sense for some positions (see Figure 2).

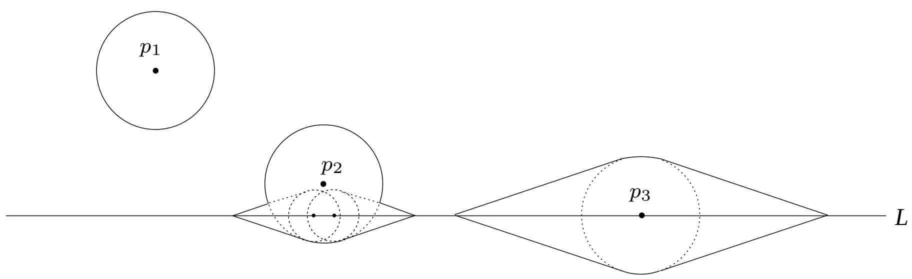

Figure 2: Unit ball for a point far away from L (p_1), a point close to L (p_2) and a point on L (p_3).

It can be proven that the time distance is a *nice metric* as defined in [4]. Hence, the Voronoi Diagram for the time metric, which we will call *Time Voronoi Diagram*, can be computed using a general sweep line algorithm [2] for Abstract Voronoi Diagrams. The only known implementation for this algorithm was written by Michael Seel in 1994 ([10]) as an extension package to LEDA [6].

In this section we show how the construction of the Time Voronoi Diagram can be drastically simplified by applying Theorem 4. If n is the number of points in S, this simplification will keep the time complexity bound of $O(n \log n)$ transforming the computation of the Time Voronoi Diagram for a set of points into the usual well known Euclidean Voronoi Diagram for some set of suitably chosen points and halflines.

Given a set of points S, we call $S_l^-(p)$ and $S_r^-(p)$ the halflines $S_l(p)$ and $S_r(p)$, respectively, for all $p \in S \cap L^+$, and $S_l^-(p)$ and $S_r^-(p)$ the halflines $S_l(p_s)$ and $S_r(p_s)$ for all $p \in S \cap L^-$. Let S_a (above) be the union of these four sets with S. We define $S_l^+(p)$, $S_r^+(p)$ and S_b (below) analogously by exchanging L^+ and L^-. We use $VR(x, X)$ for the Euclidean Voronoi Region of a point or a line $x \in X$ with respect to the set X, and $TVR(p, S)$ for the Time Voronoi Region for a point $p \in S$ with respect to the set S.

Theorem 5 *The Time Voronoi Region, $TVR(p, S)$, for a point $p \in S$ with respect to the set S is*

$$L^+ \bigcap \Big(VR(p, S_a) \cup VR(S_r^-(p), S_a) \cup VR(S_l^-(p), S_a) \Big) \quad \bigcup$$
$$L^- \bigcap \Big(VR(p, S_b) \cup VR(S_r^+(p), S_b) \cup VR(S_l^+(p), S_b) \Big).$$

(See Figure 3.)

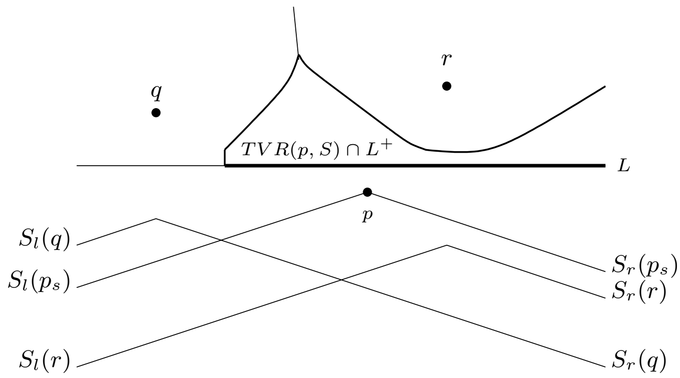

Figure 3: $TVR(p, S) \bigcap L^+ = \Big(VR(p, S_a) \cup VR(S_r(p), S_a) \cup VR(S_l(p), S_a) \Big) \bigcap L^+$.

Proof. Suppose that $p \in L^+$. If $x \in \mathbb{R}^2$ belongs to $TVR(p, S)$, then for any other site $q \in S$, with $q \neq p$, $d_t(p, x) < d_t(q, x)$. By Theorem 4, if $x \in L^+$ then

$$d_t(p, x) = \min\{d(x, p), d(x, S_r(p)), d(x, S_l(p))\}.$$

Hence, if x is in $VR(p, S_a)$, $VR(S_r(p), S_a)$ or in $VR(S_l(p), S_a)$, then $d_t(x, p) < d_t(x, q)$ for all $q \in S$, with $q \neq p$. Analogous arguments can be used to prove the cases $x \in L^-$ and $p \in L^-$. □

Both sets S_a and S_b contain $O(n)$ points and $O(n)$ halflines. In each set, these halflines can intersect in $O(n^2)$ points. Since two crossing lines produce a vertex on their Euclidean Voronoi Diagram, there are $\Omega(n^2)$ regions in the diagrams of S_a and S_b. On the other hand, since the time distance is a nice metric, the Time Voronoi Diagram contains only a linear number of regions. How can we choose in less than $O(n \log n)$ time the linear number of regions we are interested in? The main idea is that there is only a linear number of line segments in S_a (resp. S_b) whose Euclidean Voronoi Regions intersect L^+ (resp. L^-). The following two results refer to the set S_a but can be extended by symmetry to S_b.

Lemma 6 *Only the Euclidean Voronoi Regions of the line segments on the upper envelope of S_a can intersect L^+.*

Proof. By definition of S_a, all its halflines are contained in L^-. Let $p \in L^+$ be a point in the Euclidean Voronoi Region of some portion h of a halfline in S_a. Since h is the closest segment in S_a to p, then the shortest segment joining p and h cannot cross any other halfline in S_a. Furthermore, since $p \in L^+$ and $h \subseteq L^-$, then h belongs to the upper envelope of S_a. □

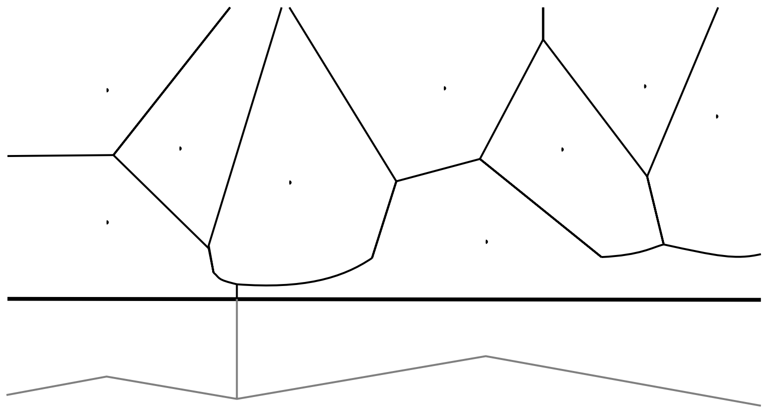

Figure 4: $TVR(p, S) \cap L^+$ for points in L^+.

Lemma 7 *The upper envelope of the halflines in S_a can be computed in $O(n \log n)$ time.*

Proof. For each point $p_i \in L^+$ (resp. $p_i \in L^-$), combine the two halflines $S_l(p_i)$ and $S_r(p_i)$ (resp. $S_l(p_{i_s})$ and $S_r(p_{i_s})$) to form a continuous function, f_i, having as domain the whole real line, our x-axis. Since all these halflines have slopes $\pm \tan(1/v)$, each pair of these functions can intersect in, at most, one point. Then, the upper envelope of S_a is

$$\min_{1 \le i \le n} f_i(x), x \in \mathbb{R}.$$

and every function can contribute with at most one piece to this upper envelope. Therefore a simple divide and conquer algorithm can be used to construct this envelope in $O(n \log n)$ time. □

As an immediate consequence of the above results and considerations we get the following result:

Theorem 8 *The Time Voronoi Diagram of a point set under the Straight Line Transportation Model can be computed in time $O(n \log n)$ via a direct specific algorithm.*

4 The v_∞ Straight Line Transportation Model

When the speed of the transportation line tends to infinity, an interesting situation arises. Let S be a set of points in the plane where no two points share the same y-coordinate. In this case, only the point which is closer to the transportation line L can benefit from the line in the Time Voronoi Diagram, since the first point reaching L, will *dominate* the whole

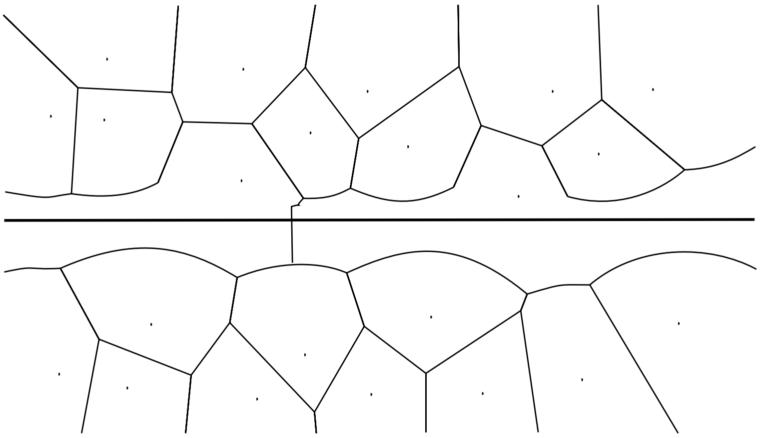

Figure 5: Time Voronoi Diagram for a set of points.

transportation line. Furthermore, there is a strong connection between this problem and Fortune's sweepline algorithm for computing the Euclidean Voronoi Diagram [3].

When v tends to infinity, the critical angle given by Snell's law of refraction, α, tends to $\pi/2$. Therefore, the function *net* is more simple to compute. Given two points in the plane p and q,

$$net(p, q) = |p_y| + |q_y|.$$

The lines $S_r(p)$ and $S_l(p)$ are parallel to L and their union is the horizontal line parallel to L at the same distance to L as p but on the opposite halfplane (see Figure 6).

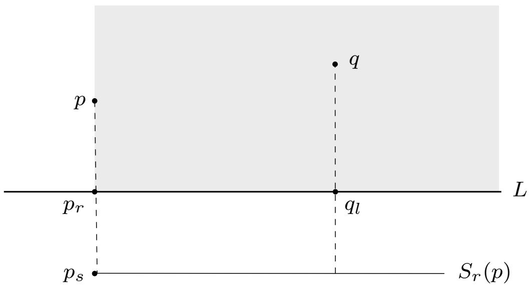

Figure 6: If v tends to infinity, then $d(q, S_r(p)) = |p_y| + |q_y|$.

Given a set of points in the plane S, if $p \in L^+$ is the closest point in S to the transportation line, then the upper envelope of S_a consists of the horizontal line in L^- at the same distance from L as p. As follows from Lemma 6, for any other point $q \in S$, with $q \neq p$, the Euclidean Voronoi Regions of the halflines associated to them are empty. Hence, p is the only point in S such that $TVR(p, S) \cap L$ is not empty.

Even though the Euclidean Voronoi Diagram for line segments is also valid for this model, there is another well-known algorithm that can be adapted to our problem and fits like a glove. Take Fortune's sweepline algorithm as described, for example in [5]. Let a horizontal sweepline move downwards and construct the Euclidean Voronoi Diagram of S step by step until the sweepline reaches $S_r(p)$, with $p \in S \cap L^+$ being the point in S closer to L. At this point, the parabolic wavefront coincides with the boundary of the Time Voronoi Region of p in L^+ and the Time Voronoi Regions of all points in $S \cap L^+$ have been constructed. An analogous construction with a sweepline moving upwards until it reaches $S_r(p_s)$ builds the Time Voronoi Diagram for L^-. This way, we can benefit from the advantages of Fortune's algorithm as

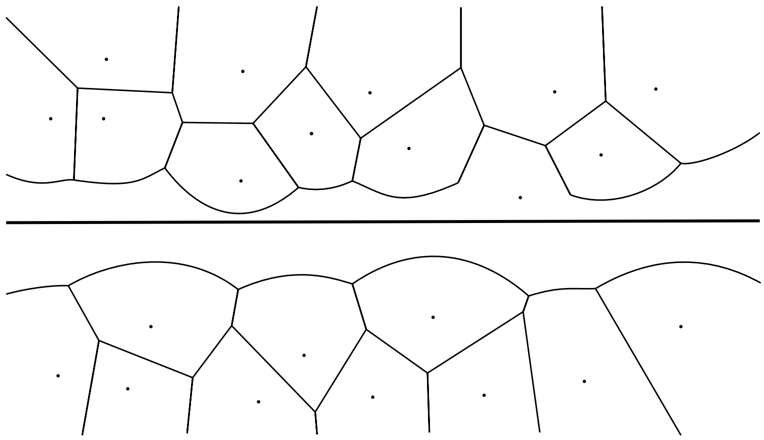

Figure 7: Time Voronoi Diagram for a set of points when v tends to infinity.

economy in space and optimal time bound in the construction of our Time Voronoi Diagram (see Figure 7), and we arrive to the following result:

Theorem 9 *The Time Voronoi Diagram of a point set under the* v_∞ *Straight Line Transportation Model can be computed in time* $O(n \log n)$ *via an adaptation of the sweepline algorithm for the standard Euclidean Voronoi Diagram.*

Acknowledgments. The authors want to thank W. Schumann for the Java implementation of the algorithm we present (Figures 4, 5 and 7 are screen captures of his applets [9]).

References

[1] F. Aurenhammer, R. Klein, *Voronoi Diagrams*, Handbook of Computational Geometry, Ed. J. Sack, J. Urrutia, pp. 201-290, Elsevier Science Publishers B.V., 2000.

[2] F. Dehne, R. Klein, *"The big sweep": On the Power of the Wavefront Approach to Voronoi Diagrams*, Algorithmica, Vol. 17, pp. 19-32, 1997.

[3] S. Fortune, *A sweep line algorithm for Voronoi Diagrams*, Algorithmica, Vol. 2, pp. 153-174, 1987.

[4] R. Klein, *Concrete and Abstract Voronoi Diagrams*, Lecture Notes in Computer Science, Springer-Verlag, 1989.

[5] R. Klein, *Algorithmische Geometrie*, Addison-Wesley-Longman, 1997.

[6] K. Mehlhorn, St. Näher, *The LEDA Platform of Combinatorial and Geometric Computing*, Cambridge University Press, 1999.

[7] A. Okabe, B. Boots, K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley and Sons, 1992.

[8] B. Palop, *Algorithmic Problems on Proximity and Location under Metric Constraints*, Ph. D. Thesis in preparation.

[9] W. Schumann, *Das Voronoi Diagramm von Liniensegmenten und eine Anwendung auf ein Problem mit zusammengesetzten Metriken*, Diplomarbeit FernUniversität Hagen, 2002.

[10] M. Seel, *Eine Implementierung abstrakter Voronoidiagramme*, Diplomarbeit U. Saarbruecken, 1994.