Hunting Voronoi Vertices in Non-Polygonal Domains*

Vincenzo Ferrucci† Mark Overmars‡ Anil Rao‡ Jules Vleugels‡

Abstract

Given three objects in the plane, a Voronoi vertex is a point that is equidistant to each under some metric. In this paper, we first consider the problem of computing a Voronoi vertex for three (possibly non-polygonal) objects in the plane under the Euclidean/Hausdorff metric. We only require the ability to query the closest point on some object from a given point; the (possibly complex) shape of the objects themselves could be unspecified. Our technique is simple, robust and iterative in nature: beginning from some initial point, it computes a sequence of points based on intermediate closest point queries. We show that this technique either converges to a Voronoi vertex or oscillates with some finite period and give conditions for each which depend on the choice of initial point and shape of the objects.

Our motivation for seeking Voronoi vertices comes from robot motion planning: Voronoi vertices are natural havens for moving robots avoiding obstacles. We conclude the paper by briefly describing an efficient implementation of a retraction-like path planner for a planar robot based on our iterative strategy for seeking Voronoi vertices.

1 Introduction

A familiar notion in computational geometry is the *Voronoi diagram* [2, 8], which can informally be defined as follows. Given a set of sites and a distance metric, the *Voronoi region* of a site is the set of points closer (under the given metric) to that site than to any other site. The Voronoi diagram is the network formed by the boundaries of the individual Voronoi regions. In the plane, this network is one-dimensional and is made up of *Voronoi edges* and *Voronoi vertices*; Voronoi vertices (edges) are the (locus of) points equidistant from the three (two) nearest sites. If the space is bounded, the Voronoi diagram is connected and preserves the connectivity of the space. The problem of computing the Voronoi diagram for a given set of sites is a familiar one in the field of computational geometry and has been extensively studied.

Robot motion planning asks for determining a collision-free motion from a start configuration to a goal configuration for a robot moving amidst but avoiding a set of obstacles. A well-known and intuitively appealing approach is to try and plan a motion that keeps the robot as far away from the obstacles as possible; this approach is often referred to as *retraction motion planning* (we refer to Latombe [10] for an overview of existing approaches). The Voronoi diagram is central to the idea of retraction motion planning. Given the Voronoi diagram in the planar configuration space of a robot, retraction motion planning works by retracting the start and goal configurations onto the diagram and then connecting them via edges and vertices of the diagram [1, 6, 12]. Whenever there exists a path, this approach is guaranteed to find one which maximizes the clearance of the robot.

In this paper our approach is similar. We compute the (configuration space) Voronoi vertices of the obstacles and find feasible paths between vertices that maintain the topology of the Voronoi diagram. While computing Voronoi vertices is well-studied for polygonal [7] and simple curved [15] obstacles, not much is known about computing them for arbitrarily curved objects. In contrast, the technique we have for computing the Voronoi vertices for three obstacles, described briefly below, does not even need the exact shape description of the obstacles. We only assume the ability to query closest points on obstacles to the current location, i.e., answers to queries of the form "given a point p and object S, determine the closest point on S from p".

*This research was partially supported by ESPRIT Basic Research Action No. 6546 (project PRoMotion) and by the Netherlands Organization for Scientific Research (NWO).

†Dipartimento di Informatica e Sistemistica, Universita di Roma "La Sapienza", Via Buonarroti 12, 00185 Roma, Italy. Email: `vincenzo@aleph.dis.uniroma1.it`

‡Department of Computer Science, Utrecht University, Padualaan 14, 3508 TB Utrecht, the Netherlands. Email: `{markov,anil,jules}@cs.ruu.nl`

Briefly, our method works as follows. Let three disjoint regular sets S_1, S_2, S_3 be given in the plane, and choose a point p. Determine three points $s_i \in S_i$ which achieve minimum distance from p. Next, compute the point equidistant from the three points s_1, s_2, s_3, and let it be q (in other words, q is the Voronoi vertex for the three point obstacles s_i). Finally, set p to q and reiterate. We observe that the sequence of points obtained in such a way often converges towards a Voronoi vertex for S_1, S_2, S_3. However, sometimes the sequence oscillates with some period. The first part of our paper (Sections 2 and 3) deals with studying the behavior of this sequence.

In Section 4 we discuss the implementation of a planar robot path planner. The approach is base on the new technique for computing Voronoi vertices which is iterative in nature, based only on nearest point computations which do not assume an exact shape description for the obstacles. Since we do not require exact shape descriptions, the method seems therefore better suited for real robotics applications than traditional Voronoi-based approaches.

Due to the page number restriction, we cannot afford to present proofs for many of our propositions. These are provided in the complete version of the paper which may be obtained by contacting the authors.

1.1 Preliminaries

Let $\mathbb{R}$ denote the set of reals, and $\mathbb{R}^2$ the plane. We include all points at infinity in $\mathbb{R}^2$. The boundary of a set S is denoted as ∂S and its interior as $int(S)$, and its convex hull as $CH(\ldots)$. The distance between two points[1] p and q is denoted as $d(p,q)$. Extend the notation to include distances between points and sets: the distance between point p and the set S is defined as $d(p,S) = \inf\{d(p,s) \mid s \in S\}$. It is clear that if $p \notin S$, $d(p,S)$ is achieved at a point $s \in \partial S$.

For a set S_i, the *Voronoi region* $Vor(S_i)$ for S_i is the set of points $\{p \mid \forall j : d(p,S_i) \leqslant d(p,S_j)\}$. For a pair of sets S_i, S_j, their bisector $bis(S_i, S_j)$ is the locus of points equidistant from both. Now let three disjoint regular[2] sets S_1, S_2, S_3 be given in the plane. A *Voronoi vertex* for S_1, S_2, S_3 is a point p such that the three distances $d(p, S_i)$ are all equal. While Voronoi regions and bisectors always exist and are uniquely defined, the set of Voronoi vertices for three sets could be empty. On the other hand, more than one Voronoi vertex could exist for three given sets. However, for three (possibly non-intersecting) convex sets, at most two Voronoi vertices can exist. Whenever our attention is focused on one Voronoi vertex for three sets, we refer to it as $Vor(S_1, S_2, S_3)$.

We define two functions ϕ and γ, and their composition ρ as follows. The function $\phi : \mathbb{R}^2 \to S_1 \times S_2 \times S_3$ maps a point $p \in \mathbb{R}^2$ to the respective closest points in the boundary of the three sets from p, i.e., $\phi(p) = (s_1, s_2, s_3)$ where $d(p, S_i)$ is achieved at $s_i \in \partial S_i$.

The function $\gamma : S_1 \times S_2 \times S_3 \to \mathbb{R}^2$ maps a triplet of points taken from the three sets to a point equidistant from the triplet (which can be at infinity, if the points are collinear). In other words, $\gamma(s_1, s_2, s_3) = q$ such that $d(q, s_1) = d(q, s_2) = d(q, s_3)$.

Our first goal in this paper is to study the behavior of the composition of γ with ϕ which we denote by $\rho : \mathbb{R}^2 \to \mathbb{R}^2$:

$$\rho = \gamma \circ \phi.$$

Specifically, we wish to investigate the relationship between $Vor(S_1, S_2, S_3)$ and $p_i = \rho^i(p_0)$ for increasing i while varying the initial point p_0 over $\mathbb{R}^2$. So the questions we may ask are:

1. For a given S_1, S_2, S_3 and p_0, does the sequence $\{p_i = \rho^i(p_0)\}$ converge to a Voronoi vertex?

   Remark: Note that a Voronoi vertex could exist at infinity. However, we still choose to use the term "convergence" to the vertex rather than "divergence" to maintain uniformity.

2. Determine necessary conditions on S_1, S_2, S_3 and p_0 under which convergence occurs.

3. Determine sufficient conditions under which the sequence oscillates.

We study these questions in Sections 2 and 3; the S_i are referred to as *objects* in these sections.

2 A Sufficient Condition for Convergence

In this section we present a sufficient condition under which the sequence $\{p_i\}$ converges to a Voronoi vertex. Other conditions for oscillations and convergence are presented in Section 3 along with some special cases.

Lemma 2.1 *Let objects S_1, S_2, S_3 and p_0 be given, and define $p_i = \rho^i(p_0)$. If $\lim_{i\to\infty} p_i$ exists, then it is a Voronoi vertex for S_1, S_2, S_3.*

This lemma implies that the sequence $\{p_i\}$ does not converge to a Voronoi vertex only if it is oscillating.[3] The sequence $\{p_i\}$ cannot exhibit chaotic behavior since our system has only two degrees of freedom [11] (Berge *et al.* [3] give a gentle introduction to dynamical systems

[1] Unless otherwise specified, we work in $\mathbb{R}^2$ and therefore a point (set of points) refers to an element (set of elements) in $\mathbb{R}^2$.

[2] A set is regular if it coincides with the closure of its interior. Formal definitions can be found in Kuratowski and Mostowski [9].

[3] Figures 1 and 2 show that oscillation can occur even for objects in general position.

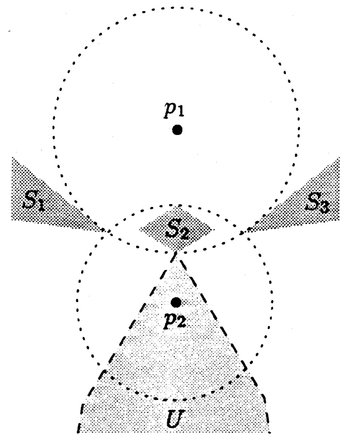

Figure 1: A case of three objects leading to oscillations in the sequence $\{p_i\}$. The S_i are the objects and U is a region bounded by segments perpendicular to the edges of the objects. For any $p_0 \in U$, p_1 and p_2 are as shown. It is easy to verify that $p_{2i} = p_2$ and $p_{2i+1} = p_1$.

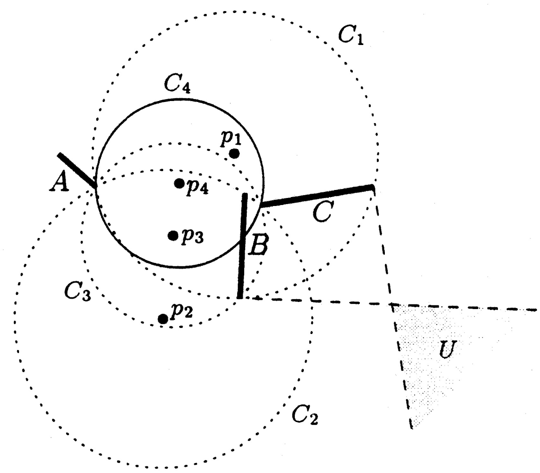

Figure 2: Another example in which oscillation occurs. A, B, C are the three line-segment objects. For any $p_0 \in U$, $p_1, \ldots, p_4$ result and thereafter $p_2, \ldots, p_4$ repeat.

and chaos). Using this lemma we will now present conditions under which the sequence converges to a Voronoi vertex. For a circle or circular disk (a circle together with its interior) C, let $rad(C)$ denote its radius and $center(C)$ its center.

Lemma 2.2 *Given a closed circular disc $D \subset \mathbb{R}^2$ and three non-collinear points $p, q, r \in D$, let C be the unique circle through p, q, r. If $center(C) \in CH(p, q, r)$ then $rad(C) \leqslant rad(D)$, and equality implies $C = \partial D$.*

In the following corollary we give a lower bound on the ratio $rad(C)/rad(D)$.

Corollary 2.1 (to Lemma 2.2) *Let r, r' be the radii of C and D, respectively. Then,*

$$\frac{r}{r'} \leqslant 1 - \left(\frac{\epsilon}{2r'}\right)^2$$

Theorem 2.1 *If $\forall i \in \mathbb{N}$, $p_{i+1} \in CH(\phi(p_i))$, then the sequence $\{p_i\}$ converges to a Voronoi vertex of S_1, S_2, S_3.*

Proof. Let the disk circumscribing the elements of $\phi(p_i)$ be denoted by D_i of radius R_i and centered at p_{i+1}. The hypothesis along with Lemma 2.2 imply that R_i is a non-increasing sequence, $R_{i+1} \leqslant R_i$ and if $R_{i+1} = R_i$, $D_i = D_{i+1}$ (the corollary to Lemma 2.2 gives an upper bound on R_{i+1}/R_i in terms of R_i). This implies that the sequence $\{D_i\}$ converges to a disk D^*. Therefore the sequence of corresponding centers $\{p_i\}$ converges to a point p^*. Lemma 2.1 implies $p^* = Vor(S_1, S_2, S_3)$. ◻

The above proof is quite general; it only relies on the definition of a Euclidean distance metric. Notice that this proof also holds for non-regular or non-compact sets S_1, S_2, S_3. It is also extendible to objects in higher dimensions (with appropriate modifications to the number of objects forming a Voronoi vertex *etc.*).

We conclude this section with examples of oscillations in $\{p_i\}$; see Figures 1 and 2. The difference between the two examples is that the initial closest points on objects are visible from each other in second example but a_1, c_1 are not visible from each other in the first. This rules out any prediction of convergence/oscillation in bases of visibility between the *initial* closest points alone. However, notice that in the second example the closest points on the obstacles from $p_1, \ldots, p_4$ are not visible from each other. This suggests there still might be a link between visibility and convergence. This is the topic of Section 3.

3 Visibility and Convergence

In this section we present some results relating the concept of visibility (as used in computational geometry) between objects to the question of convergence of the sequence $\{p_i\}$. This is motivated by the oscillation example shown in Figure 1. Notice that one of the objects "hides" a portion of the second object from a portion of the third. We assume closed convex sets for the objects in this section.

An object S_1 is said to *hide* S_2 from S_3 (and S_3 from S_2) if the set $CH(S_2 \cup S_3) \setminus int(S_1)$ is disconnected. On the other extreme, in a scene consisting of objects (S_1, S_2, S_3), objects S_2, S_3 are said to be *visible* from each other if $CH(S_2 \cup S_3) \cap S_1 = \emptyset$. Three objects are visible from each other if they are pairwise visible.

Theorem 3.1 *The following implications hold: C1 $\Leftrightarrow$ C2 $\Rightarrow$ C3, where:*

C1 *S_1, S_2, S_3 have no Voronoi vertex.*

C2 *One of S_1, S_2, S_3 hides the second from the third.*

C3 *The iterative procedure on objects S_1, S_2, S_3 produces a sequence $\{p_i\}$ that oscillates for any initial point p_0.*

This theorem gives an easy criterion from which to conclude that the sequence oscillates. The following theorem gives a similar criterion for convergence of the sequence. These criteria depend only on scene geometry and therefore may be easily verified in advance (unlike the condition in Lemma 2.2 which depends on the progress of the iterative procedure).

Theorem 3.2 *If convex objects S_1, S_2, S_3 are visible from each other, then the sequence $\{p_i\}$ converges to a Voronoi vertex from any initial point $p_0 \in \mathbb{R}^2$.*

Proof. The proof is by contradiction; we assume that S_1, S_2, S_3 are completely visible from each other yet the sequence $\{p_i\}$ starting from point p_0 oscillates with some period k. For $i \geqslant 0$, denote the triple of nearest points from p_i on S_1, S_2, S_3 as $\mathcal{T}_i = \{s_{1,i}, s_{2,i}, s_{3,i}\}$, and the circle through $\mathcal{T}_i$ with center p_{i+1} as C_i; see Figure 3. The points of $\mathcal{T}_i$ lie inside C_i because they are nearer to $center(C_{i-1})$ than the corresponding points in $\mathcal{T}_{i-1}$, and by definition they lie on C_i. For $1 \leqslant j \leqslant 3$, let $e_{j,i}$ denote the line segment joining $s_{j,i}$ with $s_{j,i+1 \bmod k}$. Since the obstacles are convex, the line segments $e_{j,i}$ are fully contained in S_j.

For $k = 2$, the line l through the intersection points of C_0 with C_1 separates $\mathcal{T}_0$ from $\mathcal{T}_1$; therefore l intersects all three obstacles. This implies that they are not completely visible.

In the general case, a case analysis shows that for any $k > 2$ there exists a similar line l that intersects $e_{j,i}$ for $1 \leqslant j \leqslant 3$, and therefore all three obstacles. Details have been omitted and can be obtained from the full paper. ◻

Although the sequence $\{p_i\}$ is not always guaranteed to converge, we can show that there always exists a "region of convergence", a maximal region such that the sequence is guaranteed to converge to a Voronoi vertex from any initial point in that region. Clearly, this region includes the target Voronoi vertex. Furthermore, these regions are of non-zero two-dimensional measure. This essentially follows from Theorem 3.2: there is a neighborhood of boundary points on each obstacle that includes the closest point from the Voronoi vertex, and is completely visible from the other two neighborhoods.

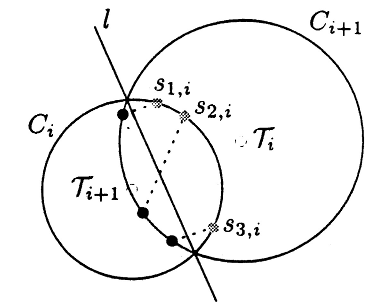

Figure 3: Illustration to the proof of Theorem 3.2, which establishes that the sequence $\{p_i\}$ converges to a Voronoi vertex if the obstacles are completely visible from each other. $\mathcal{T}_i$ denotes the triple of nearest points from p_i on the obstacles; C_i is the circle through these points. By definition, the next approximation for the Voronoi vertex p_{i+1} is the center of C_i. For oscillation with period 2, the line l intersecting all three obstacles is shown.

These neighborhoods define a subset of the plane that is of non-zero measure, includes the Voronoi vertex, and is a subset of the region of convergence. We provide a formal proof in the full paper.

This observation allows us to deal with oscillations in the sequence $\{p_i\}$ in the following way. Pick any initial point $p_0 \in \mathbb{R}^2$ and construct the corresponding sequence $\{p_i\}$. If this sequence converges, it does so to a Voronoi vertex by Lemma 2.1. Otherwise, we pick a different p_0 and construct the sequence from this point. Because the region of convergence for every Voronoi vertex is of non-zero measure, we will eventually hit a p_0 in the region of convergence. Notice that this leads to a completeness result for our technique.

4 Application to Path Planning

In this section we use the vertex-finding technique described in the previous sections to plan a path for a planar robot with two degrees of freedom that avoids a set of n static planar obstacles in its workspace. Our method is based only on nearest point computations performed in the workspace.

Going from three obstacles to a scene with several obstacles raises the following key issue. Not every triple of obstacles defines a Voronoi vertex. For example, for convex obstacles in the plane, there exists only a linear number of Voronoi vertices for the cubic number of triples [13]. Therefore, we first need to device a selection strategy that decides which triples of obstacles define a vertex.

We desire a strategy that can efficiently suggest candidate Voronoi triples. Borrowing from Overmars [14], we simply pick a point p uniformly at random over the configuration space. Next, the nearest three obstacles to p (denoted as $triple(p)$) are determined by closest-point queries; these are considered a candidate Voronoi triple. We construct the sequence $\{p_i\}$ taking $p_0 = p$, thus computing a Voronoi vertex for $triple(p)$. The approximation of the vertex for triple $\mathcal{T}$ and with seed $p_0 = p$ can be terminated when p_i and p_{i+1} are "sufficiently close", i.e., are at most some chosen parameter ϵ apart. It has been observed in practice that the convergence is very fast, rarely exceeding five iterations. This observation can be used to detect oscillations of the sequence $\{p_i\}$. If it does not converge for a relatively large number of iterations, we assume that oscillation occurs. Even if this is not true this does not affect the validity of our method, because the same triple will eventually be candidate for a different seed point. If the sequence starting from an initial point p converges to a Voronoi vertex v for $triple(p)$, we check whether this triple actually defines a vertex by determining the three obstacles nearest to v. If the triples of p and v are identical, we have found a Voronoi vertex for $triple(p)$. Otherwise, we consider $triple(v)$ a candidate triple and construct the corresponding sequence from $p_0 = v$.

To perform path planning, we build up a connectivity structure (graph) $\mathcal{G}$ around the Voronoi vertices. A path between given start and goal configurations may be searched in $\mathcal{G}$ using familiar methods. The graph is computed by incremental construction: for every Voronoi vertex v found for a triple $\mathcal{T}$, the portion of the Voronoi diagram local to v and $\mathcal{T}$ is computed as follows. Consider the portion of the Voronoi diagram given by $\mathcal{V}_{A,B} = bis(A,B) \cap \mathcal{V}$. A *via point* defined by the obstacle pair A, B is a point $p \in \mathcal{V}_{A,B}$ such that $\forall q \in \mathcal{V}_{A,B} : d(p,A) \leqslant d(q,A)$. We can uniquely associate three via points to every Voronoi vertex as follows. Suppose that v is defined by the obstacle triple $\mathcal{T} = \{A, B, C\}$; notice that v is an endpoint of $\mathcal{V}_{A,B}$ by definition. Now follow $\mathcal{V}_{A,B}$ while walking away from v until a via point is encountered. This uniquely associates a via point to $\mathcal{V}_{A,B}$. Analogously we can associate via points to $\mathcal{V}_{A,C}$ and $\mathcal{V}_{B,C}$.

The usefulness of the concept of via point lies in their use for delimiting the portion of the Voronoi diagram which is influenced by a given vertex. This makes it possible to incrementally assemble the complete diagram by appropriately joining suitable sub-diagrams. To determine the via points associated to a given Voronoi vertex v, let $\vec{d}_A(q)$ be defined as the vector from q to the point $s \in A$ which achieves minimum distance from q; obviously, $|\vec{d}_A(q)| = d(q,A)$. The vector $\vec{d}_B(q)$ is defined analogously. Now consider the sequence $\{q_i\}$ for $i = 1, 2, \ldots$, defined by:

$$q_i = q_{i-1} + \frac{\vec{d}_A(q_{i-1}) + \vec{d}_B(q_{i-1})}{2} \qquad (1)$$

Note the resemblance to the sequence $\{p_i\}$ defined in the previous section, but defined for two objects instead of three.

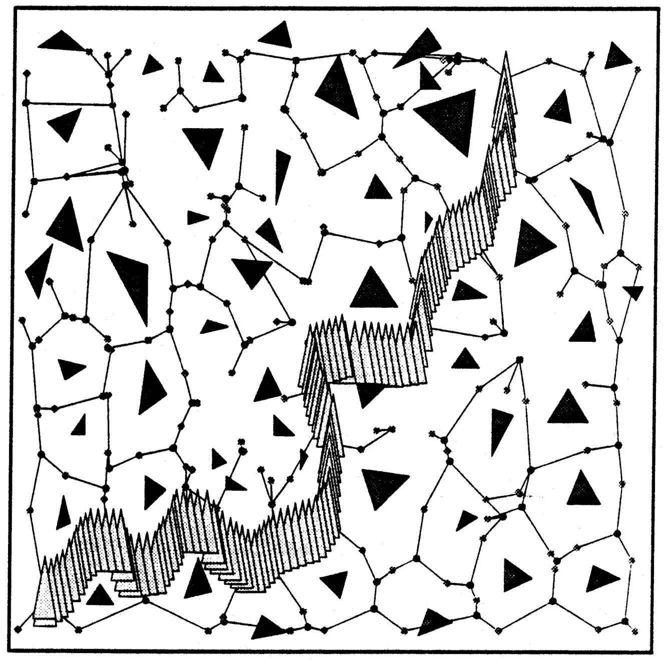

Figure 4: A set of obstacles (shown in dark grey) together with the graph representing the vertices and via points of the Voronoi diagram in the corresponding configuration space. The resulting data structure is very compact and comprises only 105 vertices and 168 via points. A collision-free path for the robot is shown in light grey.

Theorem 4.1 *For $q_0 = v$, the sequence given by Equation (1) converges to the via point p.*

The complete diagram can be incrementally obtained by appropriately joining the sub-diagrams corresponding to the portions relative to different vertices. Suppose that part of the Voronoi diagram V has been computed, and that the portion $\mathcal{V}_v$ relative to a new vertex v must be added. To correctly join $\mathcal{V}_v$ to $\mathcal{V}$ it is sufficient to keep track of the via points found so far and of their defining obstacle pairs. Let $p \in \mathcal{V}_{A,B}$ be a via point associated to v; the component $\mathcal{V}_v$ is joined to the Voronoi diagram by adding the edge from v to p to $\mathcal{V}$ (some care has to be taken if the object boundaries are parallel). The same is done for the two remaining via points associated to v.

4.1 Implementation

A motion planner for planar robots based on the method described in this paper has been implemented in C++ on a Silicon Graphics Indigo workstation. The program implements the computation of Voronoi vertices, the incremental construction of a representation of the Voronoi diagram, and the search for a collision-free path on the diagram. Testing has been based on a set of several representative scenes, each of which have their own peculiarities. One such scene, along with the diagram constructed by the program and a collision-free path for the robot is shown in Figure 4. In the full paper we include more detailed experimental results and compare the performance of our algorithm to other methods.

5 Discussion and Future Work

In this paper we have presented a simple iterative technique, based on nearest-point-on-objects queries, that converges to the Voronoi vertex for three objects or oscillates. In case of oscillations (which can be detected) and if the three objects do define a Voronoi vertex, a different choice of the initial point for the iteration is necessary for convergence. We present some conditions under which convergence is guaranteed and also conditions under which oscillations are guaranteed (the three objects do not define a vertex). However, we do not have any simultaneously necessary *and* sufficient conditions for convergence. We believe this to be an interesting open problem.

The presented conditions for convergence and oscillation are opposing: the condition for convergence being that of *complete visibility* while that for oscillation being one object *completely hides* the second from the third. We are unable to prove anything definite about intermediate cases: when there is partial visibility among the obstacles. We believe the answer will involve regions of convergence. For complete visibility, the region of convergence is the entire plane. In case of complete hiding, there is no Voronoi vertex (and therefore no region of convergence). For partial visibility, regions of convergence will constitute proper subsets of the plane. An interesting problem is to explicitly compute these regions of convergence for Voronoi vertices.

We also briefly presented an implemented motion planner for robots in a two-dimensional work space, but easily extendible to higher dimension configuration spaces. Since we only work with nearest point computations and do not need exact shape representations, perhaps our technique can be extended to changing or dynamic environments as well.

Acknowledgements

We thank Mark de Berg, Danny Halperin, Rachel Kuske, Govindan Rajeev, Otfried Schwarzkopf, and Mahendra Verma for interesting discussions. We are indebted to Otfried and his drawing editor *Ipe* for the pictures in this document, and to Geert-Jan Giezeman for his *PlaGeo* library [4], a set of geometric and graphic routines, which greatly eased our implementations.

References

[1] H. Alt and C. K. Yap. Algorithmic aspect of motion planning: a tutorial, part 2. *Algorithms Rev.*, 1(2):61–77, 1990.

[2] F. Aurenhammer. Voronoi diagrams: a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, 1991.

[3] P. Berge, Y. Pomeau, and C. Vidal. *Order within Chaos: towards a Deterministic Approach to Turbulence.* Wiley, New York, 1984.

[4] G.-J. Giezeman. *PlaGeo—A Library for Planar Geometry.* Dept. Comput. Sci., Utrecht Univ., Utrecht, the Netherlands, August 1993.

[5] B. Grünbaum. *Convex Polytopes.* Wiley, New York, NY, 1967.

[6] J. E. Hopcroft, J. T. Schwartz, and M. Sharir. *Planning, Geometry, and Complexity of Robot Motion.* Ablex Publishing, Norwood, NJ, 1987.

[7] D. Kirkpatrick and J. Snoeyink. Tentative prune-and-search for computing Voronoi vertices. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 133–142, 1993.

[8] D. G. Kirkpatrick. Efficient computation of continuous skeletons. In *Proc. 20th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 18–27, 1979.

[9] K. Kuratowski and A. Mostowski. *Set Theory.* North-Holland, Amsterdam, the Netherlands, 1976.

[10] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, USA, 1991.

[11] T.-Y. Li and J. A. Yorke. Period three implies chaos. *Amer. Math. Monthly*, 82:985–992, 1975.

[12] C. Ó'Dúnlaing, M. Sharir, and C. K. Yap. Retraction: a new approach to motion-planning. In *Proc. 15th Annu. ACM Sympos. Theory Comput.*, pages 207–220, 1983.

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

[14] M. Overmars. A random approach to motion planning. Technical Report RUU-CS-92-32, Dept. Comput. Sci., Utrecht Univ., Utrecht, the Netherlands, October 1992.

[15] C. K. Yap. An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments. *Discrete Comput. Geom.*, 2:365–393, 1987.