

Parabola separation queries and their application to stone throwing

Otfried Cheong, Hazel Everett, Hyo-Sil Kim, Sylvain Lazard, René Schott

► **To cite this version:**

Otfried Cheong, Hazel Everett, Hyo-Sil Kim, Sylvain Lazard, René Schott. Parabola separation queries and their application to stone throwing. *International Journal of Computational Geometry and Applications*, World Scientific Publishing, 2007, 17 (4), pp.349-360. <10.1142/S0218195907002379>. <inria-00434090>

HAL Id: inria-00434090

<https://hal.inria.fr/inria-00434090>

Submitted on 20 Nov 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

International Journal of Computational Geometry & Applications
© World Scientific Publishing Company

PARABOLA SEPARATION QUERIES AND THEIR APPLICATION TO STONE THROWING*

Otfried Cheong

Division of Computer Science, KAIST, Daejeon, South Korea. otfried@tclab.kaist.ac.kr

Hazel Everett

LORIA & University Nancy 2, Nancy, France. Hazel.Everett@loria.fr

Hyo-Sil Kim

Division of Computer Science, KAIST, Daejeon, South Korea. hyosil@tclab.kaist.ac.kr

Sylvain Lazard

LORIA & INRIA Lorraine, Nancy, France. Sylvain.Lazard@loria.fr

René Schott

LORIA & IECN, University Nancy 1, Nancy, France. Rene.Schott@loria.fr

Received (received date)

Revised (revised date)

Communicated by (Name)

Given two sets A and B of m non-crossing line segments in the plane, we show how to compute in $O(m \log m)$ time a data structure that uses $O(m)$ storage and supports the following query in $O(\log m)$ time: Given a parabola $\gamma : y = ax^2 + bx + c$, does γ separate A and B ? This structure can be used to build a data structure that stores a simple polygon and allows ray-shooting queries along parabolic trajectories with vertical main axis. For a polygon of complexity n , we can answer such “stone-throwing” queries in $O(\log^2 n)$ time, using $O(n \log n)$ storage and $O(n \log^2 n)$ preprocessing time. This matches the best known bound for circular ray shooting in simple polygons.

Keywords: Ray shooting; simple polygon; parabola separation queries; parabolic arc.

1. Introduction

Ray shooting is a fundamental problem in computational geometry. We are given a set of geometric objects in \mathbb{R}^d (usually $d = 2$ or 3) that we wish to preprocess and store in such a way that we can quickly answer queries of the form: Given a query ray (a half-infinite line segment), determine the first object hit (intersected)

*This research was supported by the French-Korean Science and Technology Amicable Relationships program (STAR).

by the ray. Ray shooting arises in computer graphics, in visualization, and in other geometric problems such as collision detection and motion planning.

In some applications, ray-shooting queries need to be performed along rays that are not straight. Motion planning for car-like robots, for instance, makes use of ray-shooting queries along circular arcs. In this paper, we consider another natural ray-shooting query: Which object will be hit first by a stone that moves along a parabolic trajectory under the influence of gravity?

We are aware of only one previous result that addresses ray shooting along parabolic trajectories: Sharir and Shaul⁹ very recently gave a near-linear size data structure for triangles in 3D with sublinear query time.

We concentrate here on ray shooting inside a simple polygon of complexity n . For straight rays, this problem has been solved by Hershberger and Suri,⁶ who gave a data structure that requires linear storage and answers queries in time $O(\log n)$. For circular rays, Agarwal and Sharir¹ gave a data structure achieving $O(\log^4 n)$ query time with $O(n \log^3 n)$ storage. This was improved to $O(\log^2 n)$ query time with $O(n \log n)$ storage by Cheng et al.⁴ using a novel hierarchical decomposition of simple polygons.

We make use of Cheng et al.'s hierarchical decomposition and of their framework for ray shooting. This framework guides the search for the answer to a ray-shooting query inside a simple polygon. All that remains to be done to answer parabolic ray-shooting queries is to provide a data structure that stores two sets A and B of non-crossing line segments and allows separation queries of the form: Given a parabola γ , decide whether A lies entirely *above* γ , and whether B lies entirely *below* γ . (Note that since we are interested in trajectories under the influence of gravity, our parabolas are concave and have a vertical main axis. In other words, they can be expressed in the form $y = ax^2 + bx + c$, with $a < 0$.) We will call this a *parabola separation query*.

Our result is a data structure for parabola separation queries that stores m segments in $O(m)$ storage and has query time $O(\log m)$. Plugging this data structure into Cheng et al.'s framework⁴ results in a data structure that stores a simple polygon P of complexity n in $O(n \log n)$ storage and supports ray-shooting queries along parabolic arcs originating inside P in query time $O(\log^2 n)$. These bounds equal the best known bounds for circular ray shooting. We omit a detailed description of the application of this framework; the result follows from Lemma 5 in Cheng et al.⁴

Separation queries are of interest independently of their application to ray shooting. Let A and B be sets of planar line segments. A line ℓ is a *strong separator* of A and B if all segments of A lie in one closed half-plane defined by ℓ , and all segments of B lie in the other closed half-plane defined by ℓ . Note that segments from both sets are allowed to lie on ℓ . Given A and B , a strong line separator ℓ can be found in linear time by solving a two-dimensional linear program (it suffices to ensure that the endpoints of A and B are separated). The query version of this problem is to preprocess A and B into a data structure that allows us to determine quickly whether a given line is a strong separator. This can be done by computing the

feasible region of the linear program, and preprocessing it for point location.

Our result answers the analogous question when ℓ is a parabola, albeit only for the case of parabolas with vertical main axis.

Parabola separation queries consist of two independent queries: (a) Determine whether A lies above γ ; and (b) determine whether B lies below γ . In Section 2 we give a solution for part (b) that is very similar to the solution for lines mentioned above: We simply compute the space of all feasible parabolas, and preprocess it for point location. Our solution for (a) in Section 3 is much more complicated. This is due to the fact that it does not suffice to test the parabola against the endpoints of the segments. We describe a solution based on *abstract Voronoi diagrams* as defined by Klein.⁷

2. Does B Lie Below the Parabola?

For a parabola γ given by its equation $\gamma : y = ax^2 + bx + c$ where $a < 0$, let γ^- denote the closed region lying below the parabola, that is $\gamma^- := \{(x, y) \in \mathbb{R}^2 \mid y \leq ax^2 + bx + c\}$.

We are given a set B of m line segments, which we wish to preprocess and store in a data structure such that we can answer the following query: Given a parabola γ , does B lie entirely in γ^- ?

Since γ^- is convex, a segment pq lies in γ^- if and only if both p and q lie in γ^- . It therefore suffices to test whether the set S of the $2m$ endpoints of segments in B lies in γ^- .

We represent the parabola $\gamma : y = ax^2 + bx + c$ as the point $(a, b, c) \in \mathbb{R}^3$. Each point $p_i = (x_i, y_i)$ defines a linear constraint in this space: $p_i \in \gamma^-$ if and only if $y_i \leq ax_i^2 + bx_i + c$. Since all $2m$ constraints can be written in the form $c \geq y_i - x_i^2 a - x_i b$, the set of parabolas γ with $B \subset \gamma^-$ is the region in (a, b, c) -space above the $2m$ planes $c = y_i - x_i^2 a - x_i b$.

We can now solve our problem as follows: We compute the upper envelope of these $2m$ planes in time $O(m \log m)$, project it onto the (a, b) -plane, and preprocess it for planar point location, using $O(m \log m)$ preprocessing time and $O(m)$ space.⁵ For each face of the subdivision, we store the point p_i defining the plane supporting the corresponding facet of the upper envelope. To answer a query for a parabola $\gamma : y = ax^2 + bx + c$, we locate the point (a, b) in our subdivision in time $O(\log m)$, and determine the corresponding point $p_i \in S$. We then have $B \subset \gamma^-$ if and only if $y_i \leq ax_i^2 + bx_i + c$, which we can test in constant time.

Theorem 1. *Given a set B of m line segments, we can build in time $O(m \log m)$ a data structure of size $O(m)$ that allows us to answer in $O(\log m)$ time queries of the form: Given a parabola $\gamma : y = ax^2 + bx + c$, where $a < 0$, is $B \subset \gamma^-$?*

4 *Cheong, Everett, Kim, Lazard, and Schott*

3. Does A Lie Above the Parabola?

We will make use of an entirely different parametrization of parabolas in this section. Recall that any parabola can be expressed as the locus of points equidistant from a point, the *focus*, and a line, the *directrix*. Since our parabola γ has vertical main axis, its directrix is a horizontal line $y = k$, and its focus w lies below the directrix.

We will express a parabola γ using the two parameters $k \in \mathbb{R}$ and $w \in \mathbb{R}^2$, such that $\gamma = \{p \in \mathbb{R}^2 \mid \|wp\| = |k - y_p|\}$, where $p = (x_p, y_p)$. Since the focus lies below the parabola, the closed region γ^+ lying above the parabola γ is then $\gamma^+ := \{p \in \mathbb{R}^2 \mid \|wp\| \geq |k - y_p|\}$.

We are given a set A of m non-crossing line segments,^a which we wish to preprocess and store in a data structure such that we can answer the following query: Given a parabola γ , does A lie entirely in γ^+ ? The answer to this question does not change if we replace A by its lower envelope. We start by computing this lower envelope, in time $O(m \log m)$, so that in the following we can assume that A is *x-monotone*. More precisely, we assume that any vertical line intersects at most one segment of A , or perhaps the right endpoint of one segment and the left endpoint of another segment.

For simplicity of presentation, we assume that there are no vertical segments in A that contribute to the lower envelope, and so the lower envelope does not contain isolated points. In our application to parabolic ray shooting, Cheng et al.'s hierarchical decomposition guarantees that this is indeed true. In other applications, such isolated points can easily be handled separately using the method of Section 2.

We first observe that for a point $p \in \mathbb{R}^2$, we can rewrite the condition $p \in \gamma^+$ as follows:

$$p \in \gamma^+ \Leftrightarrow \|wp\| \geq |k - y_p| \Leftrightarrow \|wp\| \geq k - y_p \Leftrightarrow \|wp\| + y_p \geq k.$$

Here we made use of the fact that if $k - y_p$ is negative, then p lies above the directrix, and therefore in γ^+ .

Let us now define a pseudo-distance function $d(u, p)$ between points u and p as follows (a geometric interpretation of this pseudo-distance will be given later):

$$d(u, p) := \|up\| + y_p.$$

From the above we find that $p \in \gamma^+$ if and only if $d(w, p) \geq k$.

Consider now a segment $s \in A$. Since s is compact and the function $p \mapsto d(u, p)$ is continuous for any $u \in \mathbb{R}^2$, the set $\{d(u, p) \mid p \in s\}$ attains its infimum in a point $p \in s$ that we denote as u_s . We define the pseudo-distance between a point u and a segment s as $d(u, s) := d(u, u_s) = \min_{p \in s} d(u, p)$. Now we observe that $s \subset \gamma^+$ if and only if for all $p \in s$ we have $d(w, p) \geq k$, which is equivalent to $d(w, s) \geq k$. Similarly, $A \subset \gamma^+$ if and only if for all $s \in A$ we have $d(w, s) \geq k$, which is equivalent to $\min_{s \in A} d(w, s) \geq k$. This implies the following lemma:

^aSegments are allowed to share endpoints, but not to intersect in any other way.

Lemma 1. *Let A be an x -monotone set of line segments, let γ be a parabola with focus w and horizontal directrix lying above w , and let $s \in A$ be a segment minimizing $d(w, s)$. Then $A \subset \gamma^+$ if and only if $s \subset \gamma^+$.*

We have thus reduced our problem to the well-known *post-office problem* (but using a somewhat unusual pseudo-distance function): We want to store our set A of line segments in such a way that we can quickly find the element $s \in A$ nearest to a given query point u (namely the focus w of the parabola). Our solution to this problem will be completely analogous to the standard solution of the Euclidean post-office problem: We will compute the Voronoi diagram of the set A (under our distance function d) and preprocess it for point-location. It remains to show that the Voronoi diagram has linear complexity, and can be computed in $O(m \log m)$ time (point-location with linear storage and $O(\log m)$ query time can then be done using standard techniques⁵).

In general, the Voronoi diagram of segments where each segment carries an additive weight is not a well-behaved Voronoi diagram: Voronoi regions can be disconnected, and the diagram can have quadratic complexity. On first sight, our problem looks even harder, as our distance function is more general: our weights vary along each segment site. Nevertheless, we will be able to show that our Voronoi diagram is well behaved, making use of the special structure of our problem: First, our set A is x -monotone. Second, our additive weights are rather special—the weight is identical to the y -coordinate of the point on the site.

We start by giving a geometric interpretation of our distance function $d(u, p)$. We observe that the Voronoi diagram is invariant under translations, so we can and will assume from now on that the set A lies entirely above the x -axis (denoted ℓ). For $p \in s$, $s \in A$, the weight y_p of p is then the distance of p from ℓ . It follows that for $u \in \mathbb{R}^2$, the distance $d(u, p)$ is the length of the shortest path from u to ℓ passing through p , see Fig. 1.

If u lies vertically above s , then clearly the shortest possible path from u to ℓ

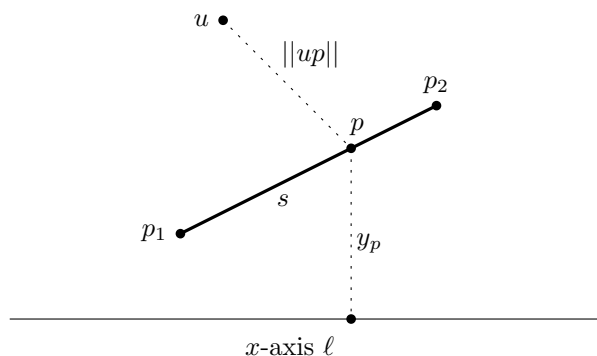
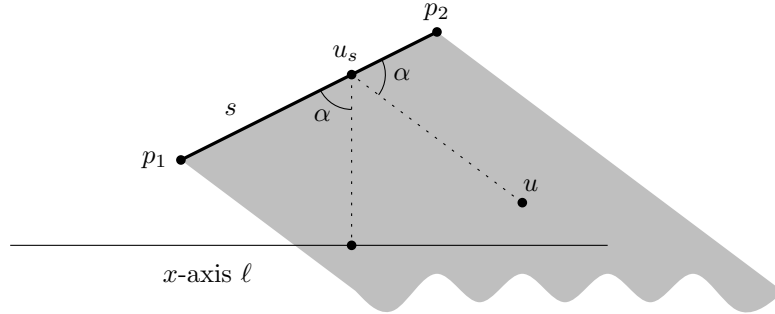


Fig. 1. Geometric interpretation of $d(u, p)$.

6 Cheong, Everett, Kim, Lazard, and Schott

Fig. 2. Fermat's principle enforces equal angles at p .

through s is a vertical segment, and so u_s lies vertically below u . On the other hand, if u lies below the supporting line of s , then the path $uu_s\ell$ enters and leaves s from below. If u_s is an interior point of s , then this path must be locally optimal, and so Fermat's principle implies that it enters and leaves s under equal angles α , see Fig. 2. The angle α is fixed by the slope of s , and so this situation arises whenever u lies in the shaded region of Fig. 2.

We can now partition the plane into five convex regions R_1, \dots, R_5 as in Fig. 3. As we observed above, for $u \in R_4$ the shortest path from u to ℓ is a vertical segment, while for $u \in R_5$ it touches s from below according to Fermat's principle. In particular, the shortest path from a point on the boundary of R_3 to ℓ goes through p_2 , and since shortest paths cannot cross, this implies that the shortest path from any point in R_3 to ℓ goes through p_2 . By the same reasoning, the shortest path from u to ℓ goes through endpoint p_1 for any point in regions R_1 and R_2 . The figure shows various points in the five regions with their shortest path to ℓ .

We observe now that in each of the five regions the distance function $d(u, s)$ can be written in a simple form. In R_1, R_2 , and R_3 , it is simply the distance to a point with an additive weight. In R_4 , it is the distance to ℓ . In R_5 , it is the distance to ℓ' , the mirror image of ℓ when reflected around the supporting line of s .

Let \prec denote some (arbitrary) total order on the segments of A , for instance the left-to-right order. For every point $u \in \mathbb{R}^2$, we let $s \prec_u s'$ if $d(u, s) < d(u, s')$ or if $d(u, s) = d(u, s')$ and $s \prec s'$. Since \prec is a total order, so is \prec_u .

We can now define the *extended Voronoi region* $EVR(s, A)$ of a segment $s \in A$ as

$$EVR(s, A) := \{u \in \mathbb{R}^2 \mid s \prec_u s' \text{ for all } s' \in A \setminus \{s\}\}. \quad (1)$$

Since \prec_u is a total order for any $u \in \mathbb{R}^2$, the extended Voronoi regions are disjoint, and $\bigcup_{s \in A} EVR(s, A) = \mathbb{R}^2$. We note that if two segments $s \prec s'$ share an endpoint p , then the points u for which $d(u, s) = d(u, s') = d(u, p)$ are part of the extended Voronoi region of s . We define the *Voronoi region* $VR(s, A)$ as the interior of $EVR(s, A)$.

Lemma 2. *Let A be an x -monotone set of line segments, and let $s \in A$. Then we have:*

- (i) *If $d(u, s) < d(u, s')$ for all $s' \in A \setminus \{s\}$, then $u \in VR(s, A)$.*
- (ii) *All interior points p of s , and all points u vertically above such an interior point $p \in s$ lie in $VR(s, A)$.*
- (iii) *For $u \in EVR(s, A)$, the segment uu_s lies entirely in $EVR(s, A)$. In particular, $u_s \in EVR(s, A)$.*
- (iv) *For $u \in VR(s, A)$, the interior of the segment uu_s lies entirely in $VR(s, A)$.*
- (v) *$EVR(s, A)$ is path-connected.*
- (vi) *$VR(s, A)$ is path-connected.*

Note that in (iv) the point u_s does not necessarily lie in $VR(s, A)$.

Proof. (i) There is a neighborhood U of u such that for all $v \in U$ we have $d(v, s) < d(v, s')$. This implies $s \prec_v s'$ for all $v \in U$, and so $U \subset EVR(s, A)$. This implies $u \in VR(s, A)$.

(ii) Let p be an interior point of s , and let u either be identical to p or lie vertically above p . Then the vertical segment from u to ℓ intersects s in p , and its length is $d(u, s)$. Since A is x -monotone and p is an interior point of s , this segment cannot touch any other segment $s' \in A$. It follows that any path from u to ℓ through a point q on another segment $s' \neq s$ cannot be straight, and is therefore longer than $d(u, s)$. This implies $d(u, s) < d(u, s')$, and by (i) we have $u \in VR(s, A)$.

(iii) Let v be a point on the segment uu_s , and assume $v \notin EVR(s, A)$. This implies that there is a segment $s' \neq s$ such that $d(v, s') \leq d(v, s)$, and, if equality

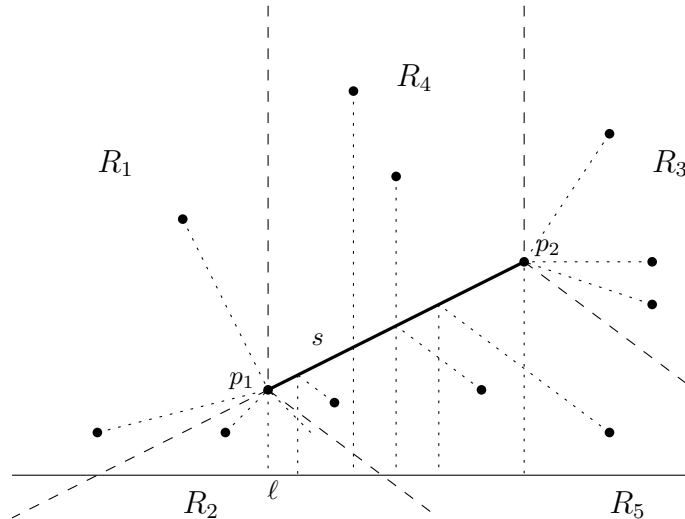


Fig. 3. The five regions with respect to s .

8 Cheong, Everett, Kim, Lazard, and Schott

holds, $s' \prec s$. Since $v_s = u_s$, $d(u, s)$ is the length of the path $uvu_s\ell$, and equal to $|uv| + d(v, s)$. On the other hand, $d(u, s')$ is at most the length of the path $uvv_s\ell$, which is $|uv| + d(v, s')$. This implies that $d(u, s') \leq d(u, s)$, and, if equality holds, $s' \prec s$. But this means $s' \prec_u s$, a contradiction to $u \in EVR(s, A)$.

(iv) Let $u \in VR(s, A) \setminus s$. There is an open neighborhood U of v that lies entirely in $VR(s, A)$ and that intersects at most two of the regions R_1, \dots, R_5 of s . By (iii), the segment vv_s lies entirely in $EVR(s, A)$ for any point $v \in U$. We note that the segment vv_s is either parallel to uu_s (if $v \in R_4$ or $v \in R_5$) or coincides at $v_s = u_s$ (if v lies in R_1, R_2 , or R_3). This implies that the union $\bigcup_{v \in U} vv_s$ contains the interior of the segment uu_s in its interior.

(v) Let u and v be in $EVR(s, A)$. By (iii), the points u_s and v_s lie in $EVR(s, A)$. Thus, by (ii), the entire segment $u_s v_s \subset EVR(s, A)$, and by (iii) again the entire path $uu_s v_s v$ lies in $EVR(s, A)$.

(vi) By (ii), the interior of s lies in $VR(s, A)$. We will show that any point $u \in VR(s, A)$ is connected to the interior of s within $VR(s, A)$. If $u_s \in VR(s, A)$, this follows immediately from (iv), so it remains to consider the case $u_s \notin VR(s, A)$. By (ii), this implies that u_s is an endpoint p of s . Without loss of generality let this be the right endpoint of s . Since $p \notin VR(s, A)$, there must be a segment $s^* \in A$ with $d(p, s^*) = d(p, s) = |p\ell|$, which implies that the left endpoint p^* of s^* must either coincide with p , or lie vertically strictly below p .

Now $u \in VR(s, A)$ implies that $d(u, p) \leq d(u, p^*)$. If p^* lies vertically strictly below p , then $d(u, p) = |up| + |pp^*| + |p^*\ell| \geq |up^*| + |p^*\ell| = d(u, p^*)$ means that $d(u, p) = d(u, p^*)$ and u lies vertically above p . But by (ii), any point vertically above p lies on the common boundary of $EVR(s, A)$ and $EVR(s^*, A)$, and not in $VR(s, A)$, a contradiction. It follows that this case cannot occur, and it remains to consider the case where p is a shared endpoint of s and s^* .

By (ii), the segment up cannot lie above $s \cup s^*$, and so it must lie below $s \cup s^*$ (see Fig 4). We pick an open neighborhood U of p such that for any $v \in U$ we have $d(v, s) < d(v, s')$ for all $s' \in A \setminus \{s, s^*\}$, and make U small enough so that the sector U' of U enclosed between s and up cannot “see” s^* . That is, for any $v \in U'$ and any point $q \in s^*$, the segment vq intersects s or up . This implies that $v \notin EVR(s^*, A)$ (as otherwise there is a contradiction with (iii)), and so $v \in EVR(s, A)$. Since this is true for all $v \in U'$, we have $U' \subset VR(s, A)$, and we can connect u to the interior

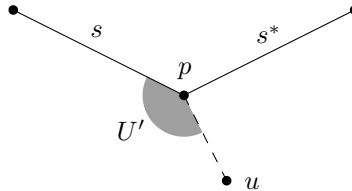


Fig. 4. Proof of Lemma 2 (vi).

of s through up and U' . \square

We now study properties of the bisector of two segments. Given s, s' in A , we define the *dominance region* $R(s, s')$ as

$$R(s, s') := \{u \in \mathbb{R}^2 \mid s \prec_u s'\}$$

We note that $R(s, s') = EVR(s, \{s, s'\})$. By the above, that implies $R(s, s') \cap R(s', s) = \emptyset$ and $R(s, s') \cup R(s', s) = \mathbb{R}^2$, and so $\partial R(s, s') = \partial R(s', s)$. We denote this boundary by $J(s, s') = J(s', s)$ and call it the *bisector* of s and s' . We let $D(s, s') := \text{int } R(s, s') = R(s, s') \setminus J(s, s')$.

Lemma 3. *For $s, s' \in A$, the bisector $J(s, s')$ is an infinite connected curve consisting of a finite number of conic arcs, which separates the unbounded path-connected regions $D(s, s')$ and $D(s', s)$.*

Proof. By the above, $J(s, s') = \partial D(s, s') = \partial D(s', s)$, and $D(s, s') \cup J(s, s') \cup D(s', s) = \mathbb{R}^2$. Without loss of generality, let $s \prec s'$.

Let R_1, \dots, R_5 be the regions of Fig. 3 for s , and let R'_1, \dots, R'_5 be the regions for s' . The intersection of each pair $R_i \cap R'_j$ is a convex polygon R_{ij} . In each region R_{ij} , the pseudo-distances to s and s' are either equal to the Euclidean distance to some line, or to the Euclidean distance to an endpoint plus an additive weight.

If s and s' share an endpoint p , then the regions R_{ij} where $d(u, s) = d(u, s') = d(u, p)$ belong entirely to $R(s, s')$ (since $s \prec s'$). In all other regions R_{ij} , we consider the set $\{u \in R_{ij} \mid d(u, s) = d(u, s')\}$. The Euclidean bisector of two lines is a pair of lines, the Euclidean bisector of two weighted points is a hyperbola, and the Euclidean bisector of a line and a weighted point is a parabola. The intersection of such a conic with the region R_{ij} is a finite number of conic arcs. For all points not on such an arc we have either $d(u, s) < d(u, s')$ or $d(u, s') < d(u, s)$, and by Lemma 2 (i) they lie in $D(s, s') = VR(s, \{s, s'\})$ or $D(s', s) = VR(s', \{s, s'\})$.

It follows that $J(s, s')$ is the finite union of conic arcs. Since $J(s, s')$ is the common boundary of $D(s, s')$ and $D(s', s)$, it follows that $J(s, s')$ consists of a finite number of infinite curves and closed loops. Now, by Lemma 2 (ii) the regions $D(s, s')$ and $D(s', s)$ are unbounded, and by Lemma 2 (vi) they are path-connected. This leaves only the possibility that $J(s, s')$ is a single infinite curve. \square

We now have all the necessary ingredients to prove that our Voronoi diagram is an abstract Voronoi diagram as defined by Klein.⁷ We recall this framework here: We are given a set A of (abstract) objects with a total order \prec . For any pair $s, s' \in A$ with $s \neq s'$, let $\hat{D}(s, s')$ be either empty or an open unbounded subset of the plane, and let $\hat{J}(s, s')$ be the boundary of $\hat{D}(s, s')$. $\hat{J}(s, s')$ is called the *bisecting curve* of s and s' . In order for the system $\{\hat{D}(s, s') \mid s, s' \in A, s \neq s'\}$ to be an *abstract Voronoi diagram*, the following conditions must hold:

- (i) $\hat{J}(s, s') = \hat{J}(s', s)$, and the regions $\hat{D}(s, s')$, $\hat{J}(s, s')$ and $\hat{D}(s', s)$ form a partition of \mathbb{R}^2 (into three disjoint sets).

10 *Cheong, Everett, Kim, Lazard, and Schott*

- (ii) If $\emptyset \neq \hat{D}(s, s') \neq \mathbb{R}^2$ then $\hat{J}(s, s')$ is homeomorphic to the open interval $(0, 1)$.
- (iii) Any two bisecting curves intersect in a finite number of connected components.

Define now $\hat{R}(s, s')$ as $\hat{D}(s, s') \cup \hat{J}(s, s')$ if $s \prec s'$, and as $\hat{D}(s, s')$ otherwise. The *extended Voronoi region* $\widehat{EVR}(s, A)$ of s is the intersection of all regions $\hat{R}(s, s')$ for $s \in A$, $s' \neq s$, and the *Voronoi region* $\widehat{VR}(s, A)$ of s is the interior of $\widehat{EVR}(s, A)$. For any non-empty subset $A' \subset A$, the Voronoi regions must satisfy the following two conditions:

- (iv) For all $s \in A'$ with $\widehat{EVR}(s, A') \neq \emptyset$ we have $\widehat{VR}(s, A') \neq \emptyset$ and both $\widehat{EVR}(s, A')$ and $\widehat{VR}(s, A')$ are path-connected.
- (v) $\mathbb{R}^2 = \bigcup_{s \in A'} \widehat{EVR}(s, A')$.

We now show that our Voronoi diagram fulfills these conditions.

Lemma 4. *Let A be an x -monotone set of line segments. Then the system of regions $D(s, s')$, $s, s' \in A$, $s \neq s'$, defines an abstract Voronoi diagram, and its Voronoi regions coincide with the regions defined earlier, that is $\widehat{EVR}(s, A) = EVR(s, A)$ and $\widehat{VR}(s, A) = VR(s, A)$.*

Proof. We set $\hat{D}(s, s') := D(s, s')$, and immediately obtain $\hat{J}(s, s') = J(s, s')$. Lemma 3 implies Klein's conditions (i), (ii), and (iii).

We need to show that $\hat{R}(s, s') = R(s, s')$. Consider a pair s, s' with $s \prec s'$, and pick a point $u \in J(s, s')$. By Lemma 2 (i) we must have $d(u, s) = d(u, s')$, and since $s \prec s'$, this means $s \prec_u s'$ and $u \in R(s, s')$. It follows that $R(s, s') = D(s, s') \cup J(s, s') = \hat{R}(s, s')$, while $R(s', s) = D(s', s) = \hat{R}(s', s)$.

We now have

$$\begin{aligned} \widehat{EVR}(s, A) &= \bigcap_{s' \in A \setminus \{s\}} \hat{R}(s, s') = \bigcap_{s' \in A \setminus \{s\}} R(s, s') = \bigcap_{s' \in A \setminus \{s\}} \{u \in \mathbb{R}^2 \mid s \prec_u s'\} \\ &= \{u \in \mathbb{R}^2 \mid s \prec_u s' \text{ for all } s' \in A \setminus \{s\}\} = EVR(s, A) \end{aligned}$$

By definition, we also have $\widehat{VR}(s, A) = VR(s, A)$. We observed before that Klein's condition (v) holds, and condition (iv) follows from Lemma 2. \square

We can now state the main result of this section.

Theorem 2. *Given a set A of m non-crossing line segments, we can build in expected time $O(m \log m)$ a data structure with $O(m)$ storage that supports in $O(\log m)$ time queries of the form: Given a parabola $\gamma : y = ax^2 + bx + c$, where $a < 0$, is $A \subset \gamma^+$?*

Proof. We first replace A by its lower envelope in time $O(m \log m)$ to obtain a set of m segments that is x -monotone. We then compute in expected time $O(m \log m)$ the Voronoi diagram of A under our distance function, using the randomized incremental algorithm by Klein et al.⁸ (or, alternatively, the deterministic algorithm of

Klein⁷). The diagram is a planar subdivision of complexity $O(m)$, and can be pre-processed in time $O(m \log m)$ using space $O(m)$ to answer point-location queries⁵ in time $O(\log m)$.

To answer a query, we locate the focus w of the parabola in the Voronoi diagram. This tells us a segment $s \in A$ such that $d(w, s) = \min_{s' \in A} d(w, s')$. By Lemma 1, it suffices to test whether s lies in γ^+ to finish the query. \square

4. Conclusions

We gave a data structure for parabola separation queries based on an abstract Voronoi diagram. Similar diagrams had been studied by Ahn et al.² and by Bae and Chwa.³

We build our data structure on a set of non-crossing line segments. This restriction is actually not necessary in Section 2, and is only used in Section 3 when we replace the input segments by their lower envelope. If the m line segments are allowed to cross, the lower envelope (and indeed the Voronoi diagram) no longer has linear complexity. The running time then becomes $O(m\alpha(m) \log m)$, where $\alpha(m)$ is the pseudo-inverse of Ackermann's function.

We then applied this data structure to the problem of shooting a parabolic ray with horizontal directrix inside a simple polygon of complexity n . The resulting data structure uses $O(n \log n)$ storage and has $O(\log^2 n)$ query time. This is probably not optimal, but matches the best known bounds for circular ray shooting in simple polygons.

It remains an interesting open problem to find an efficient solution for ray shooting along general parabolic arcs, that is, where the direction of the directrix is not known in advance. It would also be interesting to perform ray shooting along general conics. The eccentricity would become another parameter of the query.

Acknowledgment

We are grateful to an anonymous referee for a careful reading of an earlier version of this paper and pointing out that we incorrectly handled shared endpoints.

References

1. P. Agarwal and M. Sharir. Circle shooting in a simple polygon. *J. Algorithms*, 14:69–87, 1993.
2. H.-K. Ahn, O. Cheong, and R. van Oostrum. Casting a polyhedron with directional uncertainty. *Computational Geometry: Theory and Applications*, 26:129–141, 2003.
3. S. W. Bae and K.-Y. Chwa. Voronoi diagrams for a transportation network on the Euclidean plane. *International Journal of Computational Geometry & Applications*, 16:117–144, 2006.
4. S.-W. Cheng, O. Cheong, H. Everett, and R. van Oostrum. Hierarchical decompositions and circular ray shooting in simple polygons. *Discrete Comput. Geom.*, 32:401–415, 2004.

12 *Cheong, Everett, Kim, Lazard, and Schott*

5. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, Germany, 2nd edition, 2000.
6. J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *J. Algorithms*, 18:403–431, 1995.
7. R. Klein. *Concrete and Abstract Voronoi Diagrams*, volume 400 of *Lecture Notes Comput. Sci.* Springer-Verlag, 1989.
8. R. Klein, K. Mehlhorn, and S. Meiser. Randomized incremental construction of Abstract Voronoi diagrams. *Computational Geometry: Theory and Applications*, 3:157–184, 1993.
9. M. Sharir and H. Shaul. Ray shooting and stone throwing with near-linear storage. *Computational Geometry: Theory and Applications*, 30:239–252, 2005.