Parabola Separation Queries and their Application to Stone Throwing

Otfried Cheong[1], Hazel Everett[2], Hyo-Sil Kim[1],
Sylvain Lazard[2], and René Schott[3]

[1] Division of Computer Science, KAIST, Daejeon, South Korea. {otfried,hyosil}@tclab.kaist.ac.kr
[2] LORIA & IECN – INRIA Lorraine, Universities Nancy 1 & 2, Nancy, France. Firstname.Name@loria.fr

Abstract. Given two sets A and B of m non-intersecting line segments in the plane, we show how to compute in $O(m \log m)$ time a data structure that uses $O(m)$ space and allows to answer 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 with complexity n, we can answer such "stone throwing" queries in $O(\log^2 n)$ time, using $O(n \log n)$ space and $O(n \log^2 n)$ preprocessing time. This matches the best known bound for circular ray shooting in simple polygons.

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) 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 flying stone that moves under the influence of gravity along a parabolic trajectory?

We are aware of only one previous result that addresses ray shooting along parabolic trajectories: Sharir and Shaul [9] 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 space and answers queries in time $O(\log n)$. For circular rays, Agarwal and Sharir [1] gave a data structure achieving $O(\log^4 n)$ query time with $O(n \log^3 n)$ space. This was improved to $O(\log^2 n)$ query time with $O(n \log n)$ space 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 implement parabolic ray-shooting queries is to provide a data structure that stores two sets A and B of line segments and allows 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 space $O(m \log m)$ and has query time $O(\log m)$. Plugging this data structure into Cheng et al.'s framework [4] results in a data structure that stores a simple polygon P of complexity n in space $O(n \log n)$ and allows to answer 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 indepently 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 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 $\gamma^- := \{(x, y) \in \mathbb{R}^2 \mid y \leqslant ax^2 + bx + c\}$.

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 $y_i \leqslant ax_i^2 + bx_i + c$. Since all $2m$ constraints can be written in the form $c \geqslant y_i - x_i^2 a - x_i b$, the set of parabolas γ with $B \subset \gamma^-$ is the region in (a, b, c)-space above these $2m$ planes.

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 $y_i \leqslant ax_i^2 + bx_i + c$, 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 $a < 0$, does $B \subset \gamma^-$?*

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 $\gamma^+ := \{p \in \mathbb{R}^2 \mid ||wp|| \geqslant |k - y_p|\}$.

We are given a set A of m non-intersecting line segments,[1] 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 any vertical line intersects only one segment, or perhaps the right endpoint of one segment and the left endpoint of another segment. (There are no vertical segments, as they have been replaced by a point.) In other words, in the following we assume that A is *x-monotone*.

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

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

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 $d(u, p)$ with additive weight (for the point p) as follows:

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

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

Consider now a segment $s \in A$. Since s is compact and $d(u, p)$ is continuous, the set $\{d(u, p) \mid p \in s\}$ attains its minimum, and so we can define $d(u, s) := \min_{p \in s} d(u, p)$. Now we observe that $s \subset \gamma^+$ if and only if for all $p \in s$ we have $d(w, p) \geqslant k$, which is equivalent to $d(w, s) \geqslant k$. Similarly, $A \subset \gamma^+$ if and only if for all $s \in A$ we have $d(w, s) \geqslant k$, which is equivalent to $\min_{s \in A} d(w, s) \geqslant k$. It follows that we can decide whether $A \subset \gamma^+$ if we are able to compute $\min_{s \in A} d(u, s)$ for a given query point $u \in \mathbb{R}^2$ (namely the focus w of the parabola).

We have now reduced the 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. 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 techniques).

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.

Assume now that $p \in s$ is the point realizing $d(u, s)$, that is $d(u, p) = d(u, s)$. That means that p is the point in s that minimizes the length of the path $up\ell$.

If u lies vertically above s, then clearly the shortest possible path from u to ℓ through s is a vertical segment, and so p lies vertically below u. On the other hand, if u lies below the supporting line of s, then the path $up\ell$ enters and leaves s from below. If p is an interior point of s, then this path must be locally optimal, and so Fermat's law 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. For all other $u \in \mathbb{R}^2$, the shortest path from u to ℓ through s uses an endpoint of s.

We can therefore partition the plane into five convex regions $R_1, \ldots, R_5$ as in Fig. 3. In regions R_1 and R_2, the shortest path to ℓ goes through endpoint p_1, in region R_3 it goes through endpoint p_2. For $u \in R_4$,

[1] Segments are allowed to share endpoints.

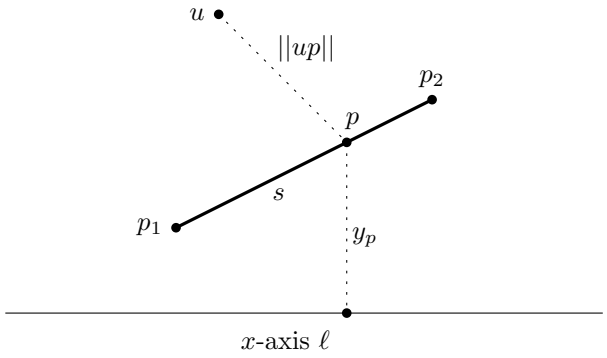

Fig. 1. Geometric interpretation of $d(u, p)$.

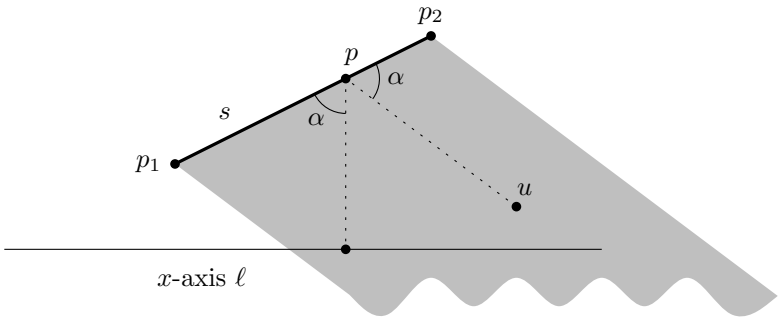

Fig. 2. Fermat's law enforces equal angles at p.

the shortest path to ℓ is a vertical segment, while for $u \in R_5$ it touches s from below according to Fermat's law. 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 $d(x, s)$ 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.

We now define the Voronoi region $VR(s, A)$ of a segment $s \in A$ as

$$VR(s, A) := \{u \in \mathbb{R}^2 \mid \forall s' \in A \text{ with } s' \neq s \text{ we have } d(u, s) < d(u, s')\}. \tag{1}$$

Lemma 1. *Let A be an x-monotone set of line segments, and let $s \in A$. Then we have:*

- *All interior points p of s, and all points u vertically above such an interior point $p \in s$ lie in $VR(s, A)$.*
- *Let $u \in VR(s, A)$, and let $p \in s$ be the point with $d(u, p) = d(u, s)$. Then the segment up lies entirely in $VR(s, A)$.*
- *$VR(s, A)$ is path-connected.*

Proof. (i) 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, and its length is $d(u, s)$. Since A is x-monotone and p is an interior point of s, this segment does not intersect any other segment $s' \in A$. It follows that any path from u to ℓ through another segment $s' \neq s$ cannot be straight, and is therefore longer than $d(u, s)$.

(ii) Let v be a point on the segment up, and assume $v \notin VR(s, A)$. This implies that there is a point $q \in s'$, $s' \neq s$, such that $d(v, q) \leqslant d(v, s) \leqslant d(v, p)$. Then the length of the path $uvq\ell$ is at most $d(u, p)$, which implies $d(u, q) \leqslant d(u, p)$, and $u \notin VR(s, A)$, a contradiction.

(iii) Follows from (i) and (ii): Let u and v be in $VR(s, A)$, and let $p, q \in s$ be their nearest points on s. Then the path $upqv$ lies in $VR(s, A)$. □

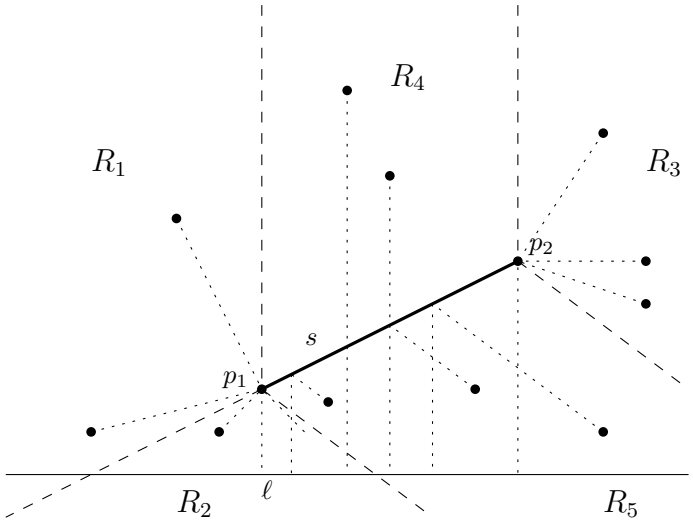

Fig. 3. The five regions with respect to s.

Consider now two segments $s, s' \in A$. We are interested in their bisector, that is, the set

$$J(s, s') := \{u \in \mathbb{R}^2 \mid d(u, s) = d(u, s')\}. \tag{2}$$

Let us also define the regions "dominated" by s and s':

$$D(s, s') := \{u \in \mathbb{R}^2 \mid d(u, s) < d(u, s')\}, \tag{3}$$

$$D(s', s) := \{u \in \mathbb{R}^2 \mid d(u, s') < d(u, s)\}. \tag{4}$$

The regions $D(s, s')$, $J(s, s')$, and $D(s', s)$ form a disjoint partition of $\mathbb{R}^2$.

Lemma 2. *Let A be an x-monotone set of line segments, and let $s, s' \in A$. Then the bisector $J(s, s')$ is an infinite simple curve consisting of at most 25 conic arcs. $J(s, s')$ partitions $\mathbb{R}^2$ into the two unbounded connected regions $D(s, s')$ and $D(s', s)$.*

Proof. We partition the plane into the five convex regions R_i for s, and similarly into the five region R'_i for s'. The intersection of each pair $R_i \cap R'_j$ is a convex polygon R_{ij}. Each R_{ij} either belongs entirely to $D(s, s')$ or $D(s', s)$, or intersects both of them. In the latter case, $R_{ij} \cap J(s, s')$ is the intersection of R_{ij} with either a line, a parabola, or a hyperbola. It follows that $J(s, s')$ is the union of at most 25 conic arcs.[2]

Applying Lemma 1 to the set $\{s, s'\}$ implies that $D(s, s') = VR(s, \{s, s'\})$ is path-connected and unbounded, and the same holds for $D(s', s)$. It follows that $J(s, s')$ is a single simple infinite curve. □

We now have all the necessary ingredients to prove that our Voronoi diagram is an Abstract Voronoi diagram as defined by Klein [7]. We start by recalling Klein's definition of Abstract Voronoi diagrams: 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 $D(s, s')$ be either empty or an open unbounded subset of the plane, and let $J(s, s')$ be the boundary of $D(s, s')$. $J(s, s')$ is called the *bisecting curve* of s and s'. The following conditions must hold:

(i) $J(s, s') = J(s', s)$, and the regions $D(s, s')$, $J(s, s')$ and $D(s', s)$ form a partition of $\mathbb{R}^2$ (into three disjoint sets).
(ii) If $\emptyset \neq D(s, s') \neq \mathbb{R}^2$ then $J(s, s')$ is homeomorphic to the open interval $(0, 1)$.

[2] Clearly, the bound 25 is far too pessimistic. For instance, R_4 and R'_4 can never intersect. We leave it to the reader to determine the exact number of pieces.

(iii) Any two bisecting curves intersect in a finite number of connected components.

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

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

Lemma 3. *Let A be an x-monotone set of line segments. Then $\{ VR(s, A) \mid s \in A \}$ is an Abstract Voronoi diagram.*

Proof. We define $D(s, s')$ as in (3). Lemma 2 implies that $D(s, s')$ is an open unbounded subset of $\mathbb{R}^2$, that $J(s, s')$ as defined in (2) is indeed its boundary, and that conditions (i), (ii), and (iii) hold.

It is easy to see that $VR(s, A')$ as defined in (1) is the interior of $EVR(s, A')$, and that $EVR(s, A')$ is a subset of the closure of $VR(s, A')$. Lemma 1 therefore implies condition (iv), and condition (v) holds trivially. □

We can now state the main result of this section.

Theorem 2. *Given a set A of m non-intersecting 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 $a < 0$, does $A \subset \gamma^+$?*

Proof. As mentioned, 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]. The diagram is a planar subdivision of complexity $O(m)$, and can be preprocessed in time $O(m \log m)$ using space $O(m)$ to answer point-location queries 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 $d(w, s) = \min_{s' \in A} d(w, s')$. It suffices to test whether s lies in γ^+ to finish the query. □

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].

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 conic. The eccentricity would become another parameter of the query.

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 with a transportation network on the Euclidean plane. In *Proc. of ISAAC*, volume 3341 of *LNCS*, pages 101–112, 2004.
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.
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*, 3:157–184, 1993.
9. M. Sharir and H. Shaul. Ray shooting and stone throwing with near-linear storage. *Computational Geometry*, 30:239–252, 2005.