

A Coreset for Approximate Furthest-Neighbor Queries in a Simple Polygon

Mark de Berg  

Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Leonidas Theocharous  

Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Abstract

Let \mathcal{P} be a simple polygon with m vertices and let P be a set of n points inside \mathcal{P} . We prove that there exists, for any $\varepsilon > 0$, a set $C \subset P$ of size $O(1/\varepsilon^2)$ such that the following holds: for any query point q inside the polygon \mathcal{P} , the geodesic distance from q to its furthest neighbor in C is at least $1 - \varepsilon$ times the geodesic distance to its further neighbor in P . Thus the set C can be used for answering ε -approximate furthest-neighbor queries with a data structure whose storage requirement is independent of the size of P . The coreset can be constructed in $O(\frac{1}{\varepsilon}(n \log(1/\varepsilon) + (n + m) \log(n + m)))$ time.

2012 ACM Subject Classification Theory of computation \rightarrow Design and analysis of algorithms

Keywords and phrases Furthest-neighbor queries, polygons, geodesic distance, coreset

Digital Object Identifier 10.4230/LIPIcs.SoCG.2024.16

Related Version *Full Version*: <https://arxiv.org/abs/2403.04513> [10]

Funding *Mark de Berg*: MdB is supported by the Dutch Research Council (NWO) through Gravitation-grant NETWORKS-024.002.003.

Leonidas Theocharous: LT is supported by the Dutch Research Council (NWO) through Gravitation-grant NETWORKS-024.002.003.

1 Introduction

Background and problem statement. In nearest-neighbor searching the goal is to preprocess a set P of n points in a metric space such that, for a query point q , one can quickly retrieve its nearest neighbor in P . Exact and approximate data structures for nearest-neighbor queries have been studied in many settings, including general metric spaces, 2- and higher-dimensional Euclidean spaces, and inside (simple) polygons.

Furthest-neighbor queries, while not as popular as nearest-neighbor queries, have been studied extensively as well. Here the goal is to report a point in P that *maximizes* its distance to the query point q . We will denote an (arbitrarily chosen, if it is not unique) furthest neighbor of q in P by $\text{FN}(q, P)$. Furthest-neighbor queries can be answered exactly by performing point location in $\text{FPVD}(P)$, the furthest-point Voronoi diagram of P . In \mathbb{R}^2 , $\text{FPVD}(P)$ has complexity $O(n)$ and it can be computed in $O(n \log n)$ time [9], and even in $O(n)$ time if the convex hull is given [4]. Furthermore, point-location queries can be answered in $O(\log n)$ time with a linear-space data structure [9], giving an optimal solution for exact furthest-neighbor queries. Furthest-site Voronoi diagrams in the plane have also been studied for non-point sites, such as line segments [5] and polygons [8], and even in an abstract setting [14].

In \mathbb{R}^d the worst-case complexity of the furthest-point Voronoi diagram is $O(n^{\lceil d/2 \rceil})$ [20]. (This also holds for polyhedral distance functions [6].) Hence, answering furthest-neighbor queries using $\text{FPVD}(P)$ leads to a data structure whose storage is exponential in the dimension d . Bespamyatnikh [7] therefore studied ε -approximate furthest-neighbor queries, where



© Mark de Berg and Leonidas Theocharous;

licensed under Creative Commons License CC-BY 4.0

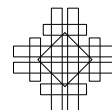
40th International Symposium on Computational Geometry (SoCG 2024).

Editors: Wolfgang Mulzer and Jeff M. Phillips; Article No. 16; pp. 16:1–16:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



the goal is to report a point in P whose distance to the query point q is at least $1 - \varepsilon$ times the distance from q to its actual furthest neighbor. He presented a data structure with $O(dn)$ storage that answers ε -approximate furthest-neighbor queries in $O(1/\varepsilon^{d-1})$ time. Note that the storage is linear in d (and n), and that the query time is independent of n . The query time is exponential in d , however, so in very high dimensions this will not be efficient. Pagh *et al.* [17], building on earlier work of Indyk [13], presented a data structure for which both the storage and the query time are linear in d : for a given $c > 1$ they can build a data structure that uses $\tilde{O}(dn^{2/c^2})$ storage and can report a c -approximate¹ furthest neighbor with constant probability in time $\tilde{O}(dn^{2/c^2})$. Finally, Goel, Indyk, and Varadarajan [11] observed that a set $C \subset P$ of at most $d + 1$ points defining the smallest enclosing ball of P has the property that, for any query point $q \in \mathbb{R}^d$, the furthest neighbor of q in C is a $\sqrt{2}$ -approximation to the actual furthest neighbor.

Our contribution. In this paper we study furthest-neighbor queries on a set P of n points inside a simple polygon \mathcal{P} with m vertices, under the geodesic distance. The *geodesic distance* between two points $x, y \in \mathcal{P}$ is the Euclidean length of the shortest path $\pi(x, y)$ connecting them in \mathcal{P} . (Shortest-path problems inside a simple polygon have received considerable attention, and it is beyond the scope of our paper to discuss all the work that has been done. The interested reader may consult the survey by Mitchell [15] for an introduction into the area.) It is known that $\text{FPVD}_{\mathcal{P}}(P)$, the geodesic furthest-point Voronoi diagram of P inside \mathcal{P} , has complexity $O(n + m)$. Wang [24] recently showed that $\text{FPVD}_{\mathcal{P}}(P)$ can be constructed in optimal $O(m + n \log n)$ time, improving an earlier result of Oh, Barba, and Ahn [16].

Since the complexity of $\text{FPVD}_{\mathcal{P}}(P)$ is $O(n + m)$ and point-location queries in the plane can be answered in logarithmic time with a linear-size data structure, the approach of answering furthest-neighbor queries using $\text{FPVD}_{\mathcal{P}}(P)$ is optimal with respect to storage and query time. This only holds for *exact* queries, however: in this paper we show that ε -approximate furthest-neighbor queries can be answered using an amount of storage that is *independent of the size of P* . More precisely, we show that there exists a set $C \subset P$ of $O(1/\varepsilon^2)$ points such that for any query point $q \in \mathcal{P}$, the geodesic furthest neighbor of q among the points in C is a $(1 - \varepsilon)$ -approximation of its actual furthest neighbor. Note that the size of the coreset is not only independent of the size of P , but also of the complexity of the polygon \mathcal{P} . We call a set C with this property an ε -coreset for furthest-neighbor queries. Using such an ε -coreset we can answer ε -approximate furthest neighbor queries with a data structure – a point-location structure on the geodesic furthest-point Voronoi diagram of C – whose storage is $O(m + 1/\varepsilon^2)$. Note that a similar result is not possible for *nearest*-neighbor queries, since for any point $p \in P$ there is a query point q such that p is the only valid answer to an ε -approximate nearest-neighbor query (even when ε is very large).

Related work. To the best of our knowledge, ε -coresets for furthest-neighbor queries have not been studied explicitly so far, although the observation by Goel, Indyk, and Varadarajan [11] mentioned above can be restated in terms of coresets: any set P in \mathbb{R}^d has a $(1/\sqrt{2})$ -coreset of size $d + 1$. It is easy to see that any set P in a space of constant doubling dimension d has a coreset of size $O(1/\varepsilon^d)$. Indeed, if r denotes the radius of the smallest enclosing ball of P , then the distance from any query point q to its furthest neighbor in P is at least r . Hence, covering B with $O(1/\varepsilon^d)$ balls of radius $\varepsilon \cdot r$ and picking an arbitrary point from P in each non-empty ball, gives an ε -coreset for furthest-neighbor queries.

¹ That is, a point $p \in P$ whose distance to the query point q is at least $1/c$ times the distance from q to its actual furthest neighbor. Note the slight inconsistency in notation when compared to the definition of ε -approximate nearest neighbor.

In \mathbb{R}^d this can be improved using ε -kernels [1]. An ε -kernel of a point set P in \mathbb{R}^d is a subset $C \subset P$ that approximates the width of P in any direction. More precisely, if $\text{width}(\phi, P)$ denotes the width of P in direction ϕ , then we require that $\text{width}(\phi, C) \geq (1 - \varepsilon) \cdot \text{width}(\phi, P)$ for any direction ϕ . It is easily checked that an $(\varepsilon/2)$ -kernel for P is, in fact, an ε -coreset for furthest-neighbor queries. Since there always exists an ε -kernel of size $O((1/\varepsilon)^{(d-1)/2})$ [1], we immediately obtain an ε -coreset for furthest-neighbor queries in \mathbb{R}^d of the same size.

Overview of our technique. While obtaining a small coreset for furthest-neighbor queries is easy under the Euclidean distance, it becomes much more challenging when P is a point set in a simple polygon \mathcal{P} and we consider the geodesic distances. It is known (and easy to see) that $\text{FN}(q, P)$ is always a vertex of $\text{RCH}(P)$, the *relative convex hull*² of P inside \mathcal{P} . Thus, one may be tempted to cover the boundary of $\text{RCH}(P)$ by geodesic balls of radius $(\varepsilon/2) \cdot \text{diam}(P)$, where $\text{diam}(P)$ denotes the diameter of P under the geodesic distance, and pick a point from P in any non-empty ball. This will give an ε -coreset since the geodesic distance from q to $\text{FN}(q, P)$ is at least $\text{diam}(P)/2$. Unfortunately, the geodesic distance does not have bounded doubling dimension, and the boundary of $\text{RCH}(P)$ can be arbitrarily much longer than $\text{diam}(P)$, the geodesic diameter of P inside \mathcal{P} . To obtain a coreset size that depends only on ε , we must therefore use a completely different approach.

The idea behind our coreset construction is as follows. Consider two points $p_1, p_2 \in P$ that define $\text{diam}(P)$ and let $\Gamma := \pi(p_1, p_2)$ be the shortest path between them. We will prove that the following property holds for any query point q inside $\text{RCH}(P)$: either there is a point $q^* \in \Gamma$ with $\text{FN}(q^*, P) = \text{FN}(q, P)$, or $\pi(q, \text{FN}(q, P))$ crosses Γ .

Query points of the former type are easy to handle: we partition Γ into $O(1/\varepsilon)$ pieces Γ_i of length $\varepsilon \cdot \text{diam}(P)$, and we add one furthest neighbor for each piece Γ_i into our coreset.

Query points of the latter type are harder to handle. Let \bar{q} denote the point where $\pi(q, \text{FN}(q, P))$ crosses Γ . The difficulty lies in the fact that $\text{FN}(\bar{q}, P)$ need not be the same as $\text{FN}(q, P)$ and, in fact, the geodesic distance from q to $\text{FN}(\bar{q}, P)$ need not even approximate the geodesic distance from q to $\text{FN}(q, P)$ well enough. The key insight is that we *can* get a good approximation if, when computing the furthest neighbor of \bar{q} , we restrict our attention to points $p \in P$ such that the angle between the first link of $\pi(\bar{q}, p)$ and the first link of $\pi(\bar{q}, \text{FN}(q, P))$ is at most ε . Thus we will define $O(1/\varepsilon)$ “cones of directions” and, for each piece Γ_i we will add one “directed furthest neighbor” for each such cone. The total size of the coreset created in this manner will be $O(1/\varepsilon^2)$.

The above does not immediately work for query points q that lie outside $\text{RCH}(P)$, because for such query points the path $\pi(q, \text{FN}(q, P))$ may not cross Γ . To overcome this, we show that there exists a set B of $O(1)$ segments, each of length at most $2\|\Gamma\|$, such that the following property holds for any query point $q \notin \text{RCH}(P)$: either $\pi(q, \text{FN}(q, P))$ crosses Γ or $\pi(q, \text{FN}(q, P))$ crosses a segment in B . We can then split B in $O(1/\varepsilon)$ pieces and then use the “cone-approach” for each piece.

2 A coreset of furthest-neighbor queries in a simple polygon

In this section we will prove our main theorem, which is as follows.

² The relative convex hull [23] of a set P of points inside a polygon \mathcal{P} is the intersection of all relatively convex sets (that is, all sets X such that for any two points $x, y \in X$ we have $\pi(x, y) \subset X$) containing P ; see Figure 1(i). Obviously the relative convex hull not only depends on P but also on the polygon \mathcal{P} , but to avoid cluttering the notation we simply write $\text{RCH}(P)$ instead of $\text{RCH}_{\mathcal{P}}(P)$.

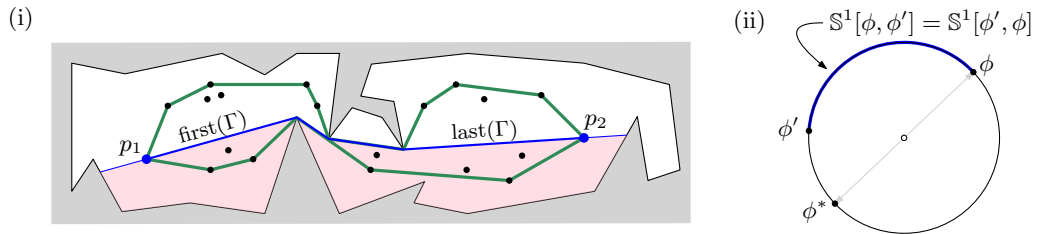
► **Theorem 1.** For any $0 < \varepsilon \leq 1$, there exists an ε -coreset $C \subset P$ of size $O(1/\varepsilon^2)$ for the furthest-neighbor problem, that is, a set C such that for any query point $q \in \mathcal{P}$ we have that $\|\pi(q, \text{FN}(q, C))\| \geq (1 - \varepsilon)\|\pi(q, \text{FN}(q, P))\|$. The coreset C can be constructed in $O(\frac{1}{\varepsilon}(n \log(1/\varepsilon) + (n + m) \log(n + m)))$ time.

For now on, we will use $\text{FN}(q)$ as a shorthand for $\text{FN}(q, P)$, the actual nearest neighbor of q . We will first prove the existence of an ε -coreset for query points that lie inside $\text{RCH}(P)$, then extend the result to query points outside $\text{RCH}(P)$, and then present an efficient algorithm to compute the coreset. Before we start we need to introduce some notation and terminology,

Notation and terminology. Recall that $\pi(x, y)$ denotes the shortest path between two points $x, y \in \mathcal{P}$. We assume that $\pi(x, y)$ is directed from x to y and we direct the edges of $\pi(x, y)$ accordingly. We denote the first edge of $\pi(x, y)$ by $\text{first}(\pi(x, y))$, while the last edge of $\pi(x, y)$ is denoted by $\text{last}(\pi(x, y))$; see Figure 1(i), where this is illustrated for Γ , the path between the points p_1 and p_2 defining the diameter. We define the *backwards extension* of a path π , denoted by $\overleftarrow{\text{ext}}(\pi)$, to be the path obtained by extending the first edge of π in backwards direction until it hits $\partial\mathcal{P}$. When we want to extend only over a given distance $\ell > 0$, then we write $\overleftarrow{\text{ext}}(\pi, \ell)$. Note that if the extension would hit $\partial\mathcal{P}$ before length ℓ is reached, then $\overleftarrow{\text{ext}}(\pi, \ell) = \overleftarrow{\text{ext}}(\pi)$. The *forward extensions* $\overrightarrow{\text{ext}}(\pi)$ and $\overrightarrow{\text{ext}}(\pi, \ell)$ are defined similarly, except that we extend the last edge of π in the forward direction. Finally, we define $\overleftrightarrow{\text{ext}}(\pi) := \overleftarrow{\text{ext}}(\pi) \cup \overrightarrow{\text{ext}}(\pi)$ and $\overleftrightarrow{\text{ext}}(\pi, \ell) := \overleftarrow{\text{ext}}(\pi, \ell) \cup \overrightarrow{\text{ext}}(\pi, \ell)$ to be the extensions of π in both directions.

As is standard in the literature, we will use the 1-sphere \mathbb{S}^1 to represent the space of all directions in \mathbb{R}^2 . Thus, for a directed line segment $s \subset \mathcal{P}$, its *direction* will be a point on \mathbb{S}^1 , which we will denote by $\text{dir}(s)$. For a shortest path $\pi(x, y) \subset \mathcal{P}$ we define its direction to be $\text{dir}(x, y) := \text{dir}(\text{first}(\pi(x, y)))$. For a point $\phi \in \mathbb{S}^1$, we will denote by ϕ^* its antipodal point. For two points $\phi, \phi' \in \mathbb{S}^1$ that are not antipodal, we denote by $\mathbb{S}^1[\phi, \phi']$ the smaller of the two arcs with endpoints ϕ, ϕ' ; see Figure 1(ii) for an example.

Recall that $\text{diam}(P)$ denotes the (geodesic) diameter of the input point set P inside \mathcal{P} . We fix two points $p_1, p_2 \in P$ that define $\text{diam}(P)$, so that $\Gamma = \pi(p_1, p_2)$. With a slight abuse of terminology we will use the term *diameter* both to refer to Γ (the actual path between p_1, p_2) and to $\text{diam}(P)$ (the length of the path). We define $\Gamma^* := \overleftrightarrow{\text{ext}}(\Gamma)$ to be the extension of Γ in both directions until $\partial\mathcal{P}$ is hit. Note that Γ^* splits \mathcal{P} into two regions, one on either side. We denote these two regions by $\mathcal{P}_1(\Gamma^*)$ and $\mathcal{P}_2(\Gamma^*)$. Note that $\mathcal{P}_1(\Gamma^*)$ – and, similarly, $\mathcal{P}_2(\Gamma^*)$ – need not be a simple polygon, because its interior need not be connected.



■ **Figure 1** (i) The boundary of $\text{RCH}(P)$, the relative convex hull, is shown in green. The points p_1, p_2 define $\text{diam}(P)$, so $\Gamma = \pi(p_1, p_2)$. The extension of Γ splits \mathcal{P} into two pieces: $\mathcal{P}_1(\Gamma^*)$, shown in pink, and $\mathcal{P}_2(\Gamma^*)$, shown in white. (ii) The 1-sphere of directions \mathbb{S}^1 .

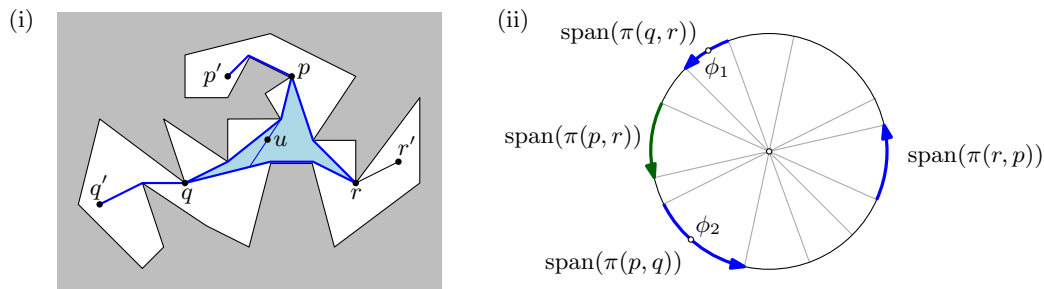


Figure 2 (i) A geodesic triangle $\Delta_{\pi} p'q'r'$ and the pseudo-triangle $\Delta_{\pi} pqr$ defined by the points where the paths meet. The extension of $\text{last}(\pi(p, u))$ hits $\pi(q, r)$. (ii) Illustration for Lemma 3.

2.1 Reduction to query points on the diameter

We start with some important properties which concern geodesic triangles. In the following, we denote by $\Delta_{\pi} p'q'r'$ the geodesic triangle with vertices $p', q', r' \in \mathcal{P}$. Let p denote the vertex where $\pi(p', q')$ and $\pi(p', r')$ split. Similarly let q denote the vertex where $\pi(q, r)$ and $\pi(q', p')$ split, and let r denote the vertex at which $\pi(r', p')$ and $\pi(r', q')$ split; see Figure 2. Note that $\Delta_{\pi} pqr$ is also a geodesic triangle. In other words, it is a (polygonal) *pseudo-triangle* [18]: a simple polygon with exactly three convex vertices, called the *corners* of the pseudo-triangle. We will need the following property, illustrated in Figure 2(i).

► **Observation 2.** *Let $\Delta_{\pi} pqr$ be a pseudo-triangle. Then for all $u \in \Delta_{\pi} pqr$ such that $u \notin \pi(q, r)$ we have that the forward extension of $\text{last}(\pi(p, u))$ will intersect $\pi(q, r)$.*

Now consider a path π . Let e_1, \dots, e_k be the ordered sequence edges of π , and let ϕ_1, \dots, ϕ_k be their directions. We define the *span* of π as $\text{span}(\pi) := \bigcup_{i=1}^{k-1} \mathbb{S}^1[\phi_i, \phi_{i+1}]$. In other words, $\text{span}(\pi)$ contains the directions into which we move while traversing π , where the “directions over which we turn” at each vertex are included. By definition, $\text{Span}(\pi)$ is connected, since any two of its consecutive arcs on \mathbb{S}^1 share an endpoint. The following lemma provides important properties about the sets of directions that occur in a pseudo-triangle. It may be known, but since we could not find a reference we provide a proof in the full version [10].

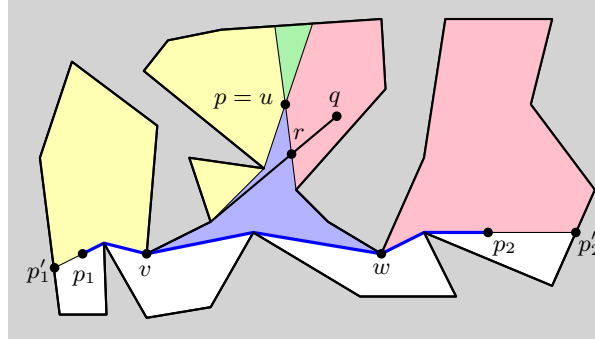
► **Lemma 3.** *Let $\Delta_{\pi} pqr$ be a pseudo-triangle. Then*

- (i) *the sets $\text{span}(\pi(p, q)), \text{span}(\pi(q, r)), \text{span}(\pi(r, p))$ are pairwise disjoint and their union contains no antipodal directions.*
- (ii) *for any $\phi_1 \in \text{span}(\pi(p, q))$ and $\phi_2 \in \text{span}(\pi(q, r))$, we have $\text{span}(\pi(p, r)) \subset \mathbb{S}^1[\phi_1, \phi_2]$.*

Finally, we need the following observation about furthest neighbors. It states that if we extend the first edge of $\pi(q, \text{FN}(q))$ backwards, and then move q along this extension, then its furthest neighbor does not change.

► **Observation 4.** *For any $q^* \in \overleftarrow{\text{ext}}(\pi(q, \text{FN}(q)))$ we have $\text{FN}(q^*) = \text{FN}(q)$.*

Proof. Intuitively, this is true because when we move q to q^* along $\overleftarrow{\text{ext}}(\pi(q, \text{FN}(q)))$, its distance to $\text{FN}(q)$ experiences the maximal possible increase (which is the full length of $\overline{q^*q}$). Formally, the observation follows if we can show that $\pi(q^*, \text{FN}(q)) = \overline{q^*q} \cup \pi(q, \text{FN}(q))$, since then $\|\pi(q^*, p)\| \leq \|\overline{q^*q}\| + \|\pi(q, p)\| \leq \|\overline{q^*q}\| + \|\pi(q, \text{FN}(q))\| = \|\pi(q^*, \text{FN}(q))\|$ for all $p \in \mathcal{P}$. It is known that any path that lies in \mathcal{P} is shortest if and only if all its internal vertices bend around reflex vertices of \mathcal{P} [3, Lemma 1]. Since the paths $\overline{q^*q} \cup \pi(q, \text{FN}(q))$ and $\pi(q, \text{FN}(q))$ have the same set of internal vertices, this concludes the proof. ◀



■ **Figure 3** The yellow region corresponds to R_1 , the blue region to R_2 , the red region to R_3 , and the green region to R_4 . In this example, the query point q lies in R_3 , so $\pi(q, p_1)$ intersects $\pi(p, p_2)$, at a point r .

We can now prove the lemma that forms the basis of our coreset construction.

► **Lemma 5.** *Let $q \in \mathcal{P}$ be a query point. Then either $\pi(q, \text{FN}(q))$ intersects Γ^* or there is a point $q^* \in \Gamma$ with $\text{FN}(q^*) = \text{FN}(q)$. Moreover, if $q \notin \text{RCH}(P)$ then $\pi(q, \text{FN}(q))$ always intersects Γ^* .*

Proof. If $q \in \Gamma$, then clearly the lemma holds, so we can assume that $q \notin \Gamma$. We now will assume that $\pi(q, \text{FN}(q))$ does not intersect Γ^* and show that then there is a point $q^* \in \Gamma$ with $\text{FN}(q^*) = \text{FN}(q)$. For the rest of the proof, let $p = \text{FN}(q)$ and assume without loss of generality that q lies in $\mathcal{P}_1(\Gamma^*)$.

Since $\pi(q, \text{FN}(q))$ does not intersect Γ^* , we have $p \in \mathcal{P}_1(\Gamma^*)$. Let u be the point where $\pi(p, p_1)$ and $\pi(p, p_2)$ split, let v be the point where $\pi(p_1, p_2)$ and $\pi(p_1, p)$ split and w the point where $\pi(p_2, p_1)$ and $\pi(p_2, p)$ split. Note that $p \in \partial \text{RCH}(P)$ and also that the angle formed between $\overrightarrow{\text{first}(\pi(p, p_1))}$, $\overrightarrow{\text{first}(\pi(p, p_2))}$ within the geodesic triangle $\Delta_{\pi} p p_1 p_2$ is convex. Now consider $\overrightarrow{\text{ext}(\pi(p_1, p))}$ and $\overrightarrow{\text{ext}(\pi(p_2, p))}$ and let $\ell_1 = \overrightarrow{\text{ext}(\pi(p_1, p))} \setminus \pi(p_1, p)$ and $\ell_2 = \overrightarrow{\text{ext}(\pi(p_2, p))} \setminus \pi(p_2, p)$. Also, let p'_1, p'_2 denote the endpoints of Γ^* . We partition $\mathcal{P}_1(\Gamma^*)$ into at most four disjoint non-empty regions, as illustrated in Figure 3.

- R_1 corresponds to the region bounded by $\pi(p'_1, p) \cup \ell_2$ and $\partial \mathcal{P}$,
- R_2 is the region enclosed by $\Delta_{\pi} u v w$,
- R_3 corresponds to the region bounded by $\pi(p'_2, p) \cup \ell_1$ and $\partial \mathcal{P}$,
- R_4 is the region bounded by $\ell_1 \cup \ell_2$ and $\partial \mathcal{P}$.

If $u \neq p$ then R_4 is empty, therefore it suffices to consider the case $u = p$. Then we have the following subcases:

- Case I: $q \in R_2$. Notice that $q \in \Delta_{\pi} u v w$ and q does not belong to $\pi(v, w)$. Therefore by Lemma 2, if $\pi(q, p)$ is extended to the direction of q , it will intersect $\pi(v, w) \subset \Gamma$ at some point q^* . By Observation 4, this point q^* has the desired property.
- Case II: $q \in R_1 \cup R_3$. Assume without loss of generality that $q \in R_3$; the case where $q \in R_1$ is symmetrical. Now $\pi(q, p_1)$ has to intersect $\pi(p, p_2)$. The reason is that R_3 is bounded by parts of shortest paths that have p_2 as an endpoint. Assume that $\pi(q, p)$ crosses $\pi(p, p_2)$ at point r . We have that

$$\|\pi(q, p)\| > \|\pi(q, p_1)\|$$

since $p = \text{FN}(q)$ and we may assume that $\|\pi(q, p)\| \neq \|\pi(q, p_1)\|$ as otherwise we could use p_1 as $\text{FN}(q)$. Also

$$\|\pi(p_1, p_2)\| \geq \|\pi(p, p_2)\|$$

since $\|\pi(p_1, p_2)\| = \text{diam}(P)$. Adding the above inequalities and using that r lies on the paths $\pi(q, p_1)$ and $\pi(p, p_2)$ we get

$$\begin{aligned} \|\pi(q, p)\| + \|\pi(p_1, p_2)\| &> \|\pi(q, r)\| + \|\pi(r, p_1)\| + \|\pi(p, r)\| + \|\pi(r, p_2)\| \\ &> (\|\pi(q, r)\| + \|\pi(r, p)\|) + (\|\pi(p_1, r)\| + \|\pi(r, p_2)\|), \end{aligned}$$

which is a contradiction, because from the triangle inequality we have that $\|\pi(q, p)\| \leq \|\pi(q, r)\| + \|\pi(r, p)\|$ and $\|\pi(p_1, p_2)\| \leq \|\pi(p_1, r)\| + \|\pi(r, p_2)\|$.

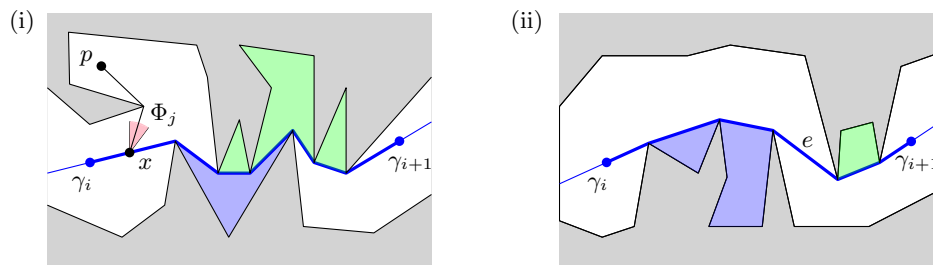
- $p \in R_4$: since the angle formed between $\text{first}(\pi(p, p_1))$ and $\text{first}(\pi(p, p_2))$ is convex, then either the angle between $\text{first}(\pi(p, q))$ and $\text{first}(\pi(p, p_1))$ is obtuse, or the angle between $\text{first}(\pi(p, q))$ and $\text{first}(\pi(p, p_2))$ is obtuse. Then by known properties of geodesic triangles [19], we have that $\|\pi(q, p)\| < \|\pi(q, p_1)\|$ or $\|\pi(q, p)\| < \|\pi(q, p_2)\|$ respectively, which contradicts that $p = \text{FN}(q)$.

So, in all cases we get a contradiction, except for Case I where we have shown the existence of a point $q^* \in \Gamma$ with $\text{FN}(q^*) = \text{FN}(q)$. Note that in Case I we have $q \in R_2$. Since $R_2 \subset \text{RCH}(P)$, this also shows that if $q \notin \text{RCH}(P)$, then $\pi(q, \text{FN}(q))$ has to intersect Γ^* . ◀

2.2 The coreset construction and its analysis

Let $k = \lceil 4\pi/\varepsilon \rceil$ and let $\phi_1, \phi_2, \dots, \phi_k$ denote k equally spaced points on \mathbb{S}^1 , which we will refer to as *canonical directions*. Note that $|\phi_{i+1} - \phi_i| \leq \varepsilon/2$ for all i . We define $\Phi_j := \mathbb{S}^1[\phi_j, \phi_{j+1}]$ and call it a *canonical cone (of directions)*. We split Γ into $\ell := \lceil 6/\varepsilon \rceil$ equal-length pieces, $\Gamma_1, \Gamma_2, \dots, \Gamma_\ell$, so that each piece Γ_i has length at most $(\varepsilon/6) \cdot \|\Gamma\|$. We denote the endpoints of a piece Γ_i by γ_i and γ_{i+1} , so $\Gamma_i = \Gamma[\gamma_i, \gamma_{i+1}]$.

A *pocket of Γ_i in $\mathcal{P}_1(\Gamma^*)$* is defined to be a simple polygonal region $R \subset \mathcal{P}_1(\Gamma^*)$ whose boundary consists of a portion of $\partial\mathcal{P}_1(\Gamma^*)$ and a portion of Γ_i . Note that the points where these two portions meet must be reflex vertices of $\partial\mathcal{P}$ whose incident edges belong to $\partial\mathcal{P}_1(\Gamma^*)$. A pocket of Γ_i in $\mathcal{P}_2(\Gamma^*)$ is defined similarly; see Figure 4. We refer to an edge of Γ_i that belongs to a pocket of Γ_i in $\mathcal{P}_1(\Gamma^*)$ or $\mathcal{P}_2(\Gamma^*)$ as a *pocket edge of Γ_i* . Any piece Γ_i has at most three non-pocket edges. Indeed, the first and last edge of Γ_i need not be pocket edges, since γ_i and γ_{i+1} need not be reflex vertices. Any other non-pocket edge must be an edge connecting two reflex vertices of \mathcal{P} , a reflex vertex u from $\mathcal{P}_1(\Gamma^*) \cap \partial\mathcal{P}$ and a reflex vertex v from $\mathcal{P}_2(\Gamma^*) \cap \partial\mathcal{P}$. Moreover, for uv to be a non-pocket edge, all vertices of Γ_i that lie behind v (as seen from u) must also be reflex vertices of $\mathcal{P}_2(\Gamma^*) \cap \partial\mathcal{P}$, and all vertices of Γ_i that lie behind u (as seen from v) must also be reflex vertices of $\mathcal{P}_1(\Gamma^*) \cap \partial\mathcal{P}$. Hence, there can be at most one such edge, which we will refer to as an *intermediate edge of Γ_i* . See Figure 4(ii) for an illustration.



■ **Figure 4** Pockets in $\mathcal{P}_1(\Gamma^*)$, which is the region below Γ^* , are shown in blue and pockets in $\mathcal{P}_2(\Gamma^*)$ are shown in green. (i) The point p is reachable from x in direction of the cones Φ_j , which is shown in pink. (ii) the edge e is an intermediate edge.

For a point $p \in P$, a piece Γ_i and a canonical cone Φ_j , we will say that p is *reachable from Γ_i in the cone Φ_j* , if there exists a point $x \in \Gamma_i$ such that $\text{dir}(x, p) \in \Phi_j$ and $\pi(x, p) \cap \Gamma_i = x$; see Figure 4(i). In other words, there is a point $x \in \Gamma_i$ such that the shortest path $\pi(x, p)$ immediately leaves Γ_i and the direction of the first edge of $\pi(x, p)$ lies in the cone Φ_j . We may then just say that p is reachable from Γ_i , or that p is reachable from the edge of Γ_i that x belongs to. Now we are ready to define the points placed in our coreset for each Γ_i .

Coreset definition. For each piece Γ_i we will put a set C_i of points into the coreset. The final coreset C is then defined as $C = \bigcup_i C_i$. The points placed into C_i will be of five different types, as explained next.

- If Γ_i has an intermediate edge e , let E_1, E_2 denote the two half-polygons that \mathcal{P} is split into by e . Then we start by placing in C_i the points α_i, α'_i defined as follows: α_i is the point in $P \cap E_1$ which is furthest from γ_i over all points $p \in P \cap E_1$. Similarly, α'_i is the point furthest from γ_i over all points $p \in P \cap E_2$.

Note that for the points α_i, α'_i we do not care whether they lie in $\mathcal{P}_1(\Gamma^*)$ or $\mathcal{P}_2(\Gamma^*)$. We will also place in C_i some points of P that must belong to $\mathcal{P}_1(\Gamma^*)$, and some points of P that must belong to $\mathcal{P}_2(\Gamma^*)$. We denote these two sets of points by $C_i(\mathcal{P}_1(\Gamma^*))$ and $C_i(\mathcal{P}_2(\Gamma^*))$ respectively. We then have $C_i = C_i(\mathcal{P}_1(\Gamma^*)) \cup C_i(\mathcal{P}_2(\Gamma^*)) \cup \{\alpha_i, \alpha'_i\}$. We now explain which points are placed in $C_i(\mathcal{P}_1(\Gamma^*))$. The set $C_i(\mathcal{P}_2(\Gamma^*))$ is defined analogously.

- We place in $C_i(\mathcal{P}_1(\Gamma^*))$ the point $r_i \in P \cap \mathcal{P}_1(\Gamma^*)$ that is furthest from γ_i over all points $p \in P \cap \mathcal{P}_1(\Gamma^*)$.
- Let $X_i \subset P$ denote the points that lie in a pocket of Γ_i . Then we place in $C_i(\mathcal{P}_1(\Gamma^*))$ the point $x_i \in X_i$ that is furthest from γ_i over all points $p \in X_i \cap \mathcal{P}_1(\Gamma^*)$. Note this is not done for each pocket separately: we only pick a single point over all pockets.
- Let $F_{i,j}$ denote the set of points that are reachable from $\text{first}(\Gamma_i)$ in the cone Φ_j . For each set $F_{i,j}$ we place in $C_i(\mathcal{P}_1(\Gamma^*))$ the point $f_{i,j}$ defined as follows: $f_{i,j}$ is the point in $F_{i,j} \cap \mathcal{P}_1(\Gamma^*)$ that is furthest away from γ_i .
- Let $L_{i,j}$ denote the set of points that are reachable from $\text{last}(\Gamma_i)$ in the cone Φ_j . For each set $L_{i,j}$ we place in $C_i(\mathcal{P}_1(\Gamma^*))$ the point $\ell_{i,j}$ defined as follows: $\ell_{i,j}$ is the point in $L_{i,j} \cap \mathcal{P}_1(\Gamma^*)$ that is furthest away from γ_i .

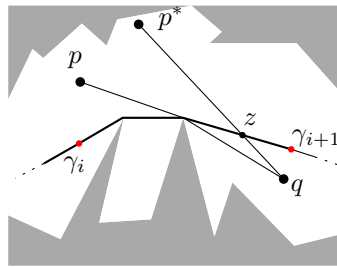
Proof of Theorem 1 when the query point lies inside $\text{RCH}(P)$. Without loss of generality, we assume that $q \in \mathcal{P}_2(\Gamma^*)$. For brevity, we define $p := \text{FN}(q)$ to be the furthest neighbor q . The following lemma will be used repeatedly in our proof.

► **Lemma 6.** *Let $p^* \in C$ be such that $\pi(q, p^*)$ intersects Γ_i , and suppose $\|\pi(\gamma_i, p^*)\| \geq \|\pi(\gamma_i, p)\|$. Then $\|\pi(q, p^*)\| \geq (1 - \varepsilon) \cdot \|\pi(q, p)\|$. Hence, p^* is a good approximation for $\text{FN}(q)$.*

Proof. Let z be a point where $\pi(q, p^*)$ intersects Γ_i ; see Figure 5. Then we have

$$\begin{aligned}
 \|\pi(q, p)\| &\leq \|\pi(q, z)\| + \|\pi(z, \gamma_i)\| + \|\pi(\gamma_i, p)\| && \text{(Triangle Inequality)} \\
 &\leq \|\pi(q, z)\| + \|\pi(z, \gamma_i)\| + \|\pi(\gamma_i, p^*)\| && \text{(by assumption)} \\
 &\leq 2\|\pi(z, \gamma_i)\| + \|\pi(q, z)\| + \|\pi(z, p^*)\| && \text{(Triangle Inequality)} \\
 &\leq 2\|\Gamma_i\| + \|\pi(q, p^*)\| && \text{(since } \|\pi(z, \gamma_i)\| \leq \|\Gamma_i\| \text{ and } z \in \pi(q, p^*)\text{)} \\
 &< \frac{\varepsilon}{2} \cdot \|\Gamma\| + \|\pi(q, p^*)\| && \text{(since } \|\Gamma_i\| \leq \frac{\varepsilon}{6} \cdot \|\Gamma\|\text{)} \\
 &\leq \varepsilon \cdot \|\pi(q, p)\| + \|\pi(q, p^*)\| && \text{(since } \|\Gamma\| \leq 2\|\pi(q, \text{FN}(q))\| \text{ for any } q \in \mathcal{P}\text{)}
 \end{aligned}$$

and so indeed $\|\pi(q, p^*)\| \geq (1 - \varepsilon) \cdot \|\pi(q, p)\|$. ◀



■ **Figure 5** Illustration for the proof of Lemma 6.

If $p \in \{p_1, p_2\}$, then Theorem 1 trivially holds since $\{p_1, p_2\} \subset C$. So now we assume that $p \notin \{p_1, p_2\}$. Lemma 5 gives us that either $\pi(q, p)$ crosses Γ^* or the backwards extension of $\pi(q, p)$ will intersect Γ . Note that in the former case $\pi(q, \text{FN}(q))$ actually intersects Γ , since $q \in \text{RCH}(P)$ and shortest paths between points inside $\text{RCH}(P)$ do not leave $\text{RCH}(P)$. We first consider the latter case, where the backwards extension of $\pi(q, p)$ intersects Γ .

► **Lemma 7.** *If the backwards extension of $\pi(q, p)$ intersects Γ , then there is a point $p^* \in C$ such that $\|\pi(q, p^*)\| \geq (1 - \varepsilon) \cdot \|\pi(q, p)\|$.*

Proof. Let x be the point where the backwards extension of $\pi(q, p)$ hits Γ and let Γ_i be the piece containing x . Let $r_i^* \in C$ be the point reachable from Γ_i which is furthest from γ_i over all points $p \in P \cap \mathcal{P}_2(\Gamma^*)$ and that was put into the coreset C . Thus the definition of r_i implies $\|\pi(\gamma_i, r_i^*)\| \geq \|\pi(\gamma_i, p)\|$. We now have

$$\begin{aligned}
 \|\pi(q, p)\| &= \|\pi(x, p)\| - \|\pi(q, x)\| \quad (\text{since } x \text{ is on the backwards extension}) \\
 &\leq \|\pi(x, \gamma_i)\| + \|\pi(\gamma_i, p)\| - \|\pi(q, x)\| \quad (\text{Triangle Inequality}) \\
 &\leq \|\pi(x, \gamma_i)\| + \|\pi(\gamma_i, r_i^*)\| - \|\pi(q, x)\| \quad (\text{definition of } r_i^*) \\
 &\leq 2\|\pi(x, \gamma_i)\| + \|\pi(x, r_i^*)\| - \|\pi(x, q)\| \quad (\text{Triangle Inequality}) \\
 &\leq 2\|\Gamma_i\| + \|\pi(x, r_i^*)\| - \|\pi(x, q)\| \quad (\text{since } x \in \Gamma_i \text{ and } \gamma_i \in \Gamma_i) \\
 &\leq 2\|\Gamma_i\| + \|\pi(q, r_i^*)\| \quad (\text{Triangle Inequality}) \\
 &< \varepsilon\|\pi(q, p)\| + \|\pi(q, r_i^*)\| \quad (\text{since } \|\Gamma_i\| \leq \frac{\varepsilon}{6}\|\Gamma\| < \frac{\varepsilon}{2}\|\pi(q, p)\|)
 \end{aligned}$$

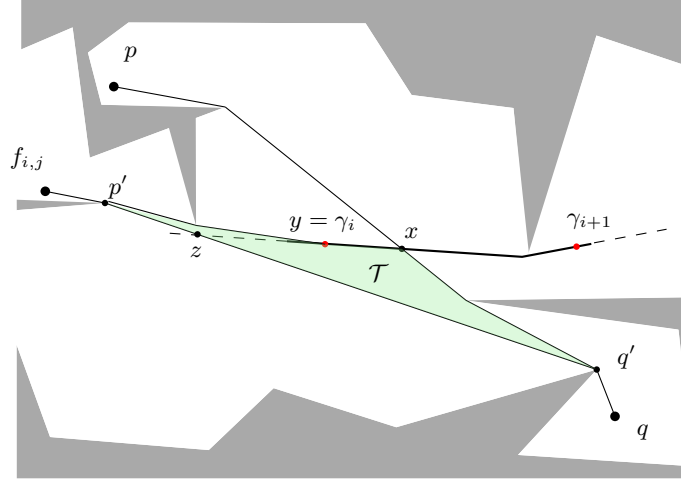
and so $\|\pi(q, r_i^*)\| \geq (1 - \varepsilon) \cdot \|\pi(q, p)\|$. By taking $p^* := r_i^*$ we thus obtain a point satisfying the conditions of the lemma. ◀

We are left with the case where $\pi(q, p)$ crosses Γ_i , which is handled by the next two lemmas.

► **Lemma 8.** *If $\pi(q, p)$ crosses Γ_i at a pocket edge or an intermediate edge, then there is a point $p^* \in C$ such that $\|\pi(q, p^*)\| \geq (1 - \varepsilon) \cdot \|\pi(q, p)\|$.*

Proof. Since we assumed that $q \in \mathcal{P}_2(\Gamma^*)$, we have $p \in \mathcal{P}_1(\Gamma^*)$. If $\pi(q, p)$ crosses Γ_i at a pocket edge, then either q or p lies in a pocket of Γ_i . Recall that C contains the point $x_i \in X_i \cap \mathcal{P}_1(\Gamma^*)$ that is furthest away from γ_i , where X_i denotes the set of points in a pocket of Γ_i . Note that $\pi(q, x_i)$ must cross Γ_i . If p lies in a pocket then $\|\pi(\gamma_i, x_i)\| \geq \|\pi(\gamma_i, p)\|$ by definition of x_i , and so we can apply Lemma 6 with $p^* := x_i$. If q lies in a pocket then $\|\pi(\gamma_i, r_i)\| \geq \|\pi(\gamma_i, p)\|$ by definition of r_i , and so we can apply Lemma 6 with $p^* := r_i$.

If $\pi(q, p)$ crosses Γ_i at its intermediate edge e , then for the point $\alpha_i \in C$ it has to be that $\pi(q, \alpha_i)$ crosses e . Hence, we can apply Lemma 6 with $p^* := \alpha_i$. Thus in both cases the point p^* has the desired property. ◀



■ **Figure 6** Illustration for Subcase II.

It remains to deal with the most technically involved case, which is when $\pi(q, p)$ crosses $\text{first}(\Gamma_i)$ or $\text{last}(\Gamma_i)$. In the following proof, we denote by $\square_{\pi}abcd$ the geodesic quadrilateral with vertices $a, b, c, d \in \mathcal{P}$.

► **Lemma 9.** *If $\pi(q, p)$ crosses $\text{first}(\Gamma_i)$ or $\text{last}(\Gamma_i)$ then there is a point $p^* \in C$ such that $\|\pi(q, p^*)\| \geq (1 - \varepsilon) \cdot \|\pi(q, p)\|$.*

Proof. Assume that $\pi(q, p)$ crosses $\text{first}(\Gamma_i)$ – the case that it crosses $\text{last}(\Gamma_i)$ is treated similarly – and let x be the crossing point. Let Φ_j be the canonical cone such that $\text{dir}(x, p) \in \Phi_j$. We will show that $f_{i,j} \in C$ is a good approximation of the furthest neighbor of q , so setting $p^* := f_{i,j}$ will prove the lemma. Let $y \in \text{first}(\Gamma_i)$ be a point from which $f_{i,j}$ is reachable. We will consider three subcases.

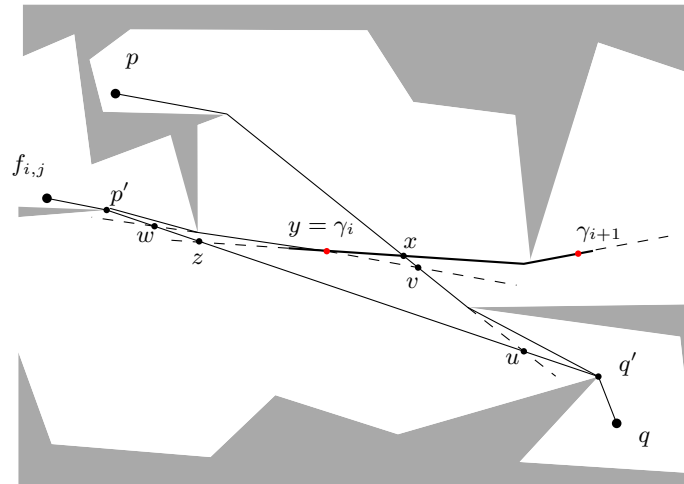
Subcase I: $\pi(q, f_{i,j})$ crosses Γ_i . By definition of $f_{i,j}$ we have $\|\pi(\gamma_i, f_{i,j})\| \geq \|\pi(\gamma_i, p)\|$, and so it follows immediately from Lemma 6 that $\|\pi(q, f_{i,j})\| \geq (1 - \varepsilon) \cdot \|\pi(q, p)\|$.

To define the other two subcases, let q' denote the point where the paths $\pi(q, x)$ and $\pi(q, f_{i,j})$ split, and let p' denote the point where the paths $\pi(f_{i,j}, q)$ and $\pi(f_{i,j}, y)$ split; see Figure 6. The two remaining subcases depend on whether $\text{dir}(q', x)$ is very different from $\text{dir}(x, p)$ or, more precisely, depending on whether $\text{dir}(q', x)$ lies in Φ_j or not.

Subcase II: $\pi(q, f_{i,j})$ does not cross Γ_i and $\text{dir}(q', x) \in \Phi_j$. Since also $\text{dir}(x, p) \in \Phi_j$, we know that $\text{dir}(e) \in \Phi_j$ for every edge $e \in \pi(q', x)$. Let z denote the point where $\pi(q, f_{i,j})$ crosses Γ , and note that we assumed $z \notin \Gamma_i$. An illustration is shown in Figure 6, where for ease of illustration we assumed that $y = \gamma_i$; this does not make any difference in the proof.

Consider the geodesic quadrilateral $\square_{\pi}q'xyp'$. Then the four shortest paths forming the boundary of this quadrilateral are disjoint, except for shared endpoints. Since the quadrilateral is fully contained in \mathcal{P} (because \mathcal{P} is simple), all interior vertices of these paths are reflex vertices of $\square_{\pi}q'xyp'$. Moreover, the following holds: if x lies between z and y on Γ^* then in $\square_{\pi}q'xyp'$ the angle at x is reflex. Similarly, if y lies between z and x on Γ^* (as in Figure 6) then the angle at y is reflex. We will assume from now on without loss of generality that y lies between z and x .

We will first show that the edges of $\pi(q', p')$ all have a direction in the canonical cone Φ_j .



■ **Figure 7** Illustration for Subcase II.

▷ **Claim 10.** For every edge $e \in \pi(q', p')$, we have that $\text{dir}(e) \in \Phi_j$.

Proof. Since in $\square_{\pi} q' x y p'$ the angle at y is reflex, $\square_{\pi} q' x y p'$ is a pseudo-triangle \mathcal{T} with sides $\pi(q', x)$, and $\pi(x, y) \cup \pi(y, p')$, and $\pi(q', p')$. Let e_1 be the last edge of $\pi(q', x)$. Then we know that $\text{dir}(e_1) \in \Phi_j$ and $\text{dir}(y, p') \in \Phi_j$. Therefore, we have that $\mathbb{S}^1[\text{dir}(e_1), \text{dir}(y, p')] \subset \Phi_j$. Since $\text{dir}(e_1) \in \text{span}(q', x)$ and $\text{dir}(y, p') \in \text{span}(x, p')$, Lemma 3(ii) gives us that $\text{span}(\pi(q', p')) \subset \mathbb{S}^1[\text{dir}(e_1), \text{dir}(y, p')] \subset \Phi_j$, which is what we wanted to prove. ◁

Claim 10 implies the following.

▷ **Claim 11.** $\|\pi(q', x)\| + \|\pi(y, p')\| < (1 + \frac{\varepsilon}{2})\|\pi(q', p')\| + \|\pi(x, y)\|$

Proof. We extend e_1 to the interior of \mathcal{T} until it intersects $\pi(q', p')$ at a point u . We also extend $\text{first}(\pi(y, p'))$ to both directions until it intersects \overline{xu} at a point v and $\pi(q', p')$ at a point w . Refer to Figure 7 for an illustration, where (to reduce the clutter in the figure) point w coincides with p' .

Then, by repeated application of the triangle inequality we have:

$$\begin{aligned} \|\pi(q', x)\| + \|\pi(y, p')\| &\leq \|\pi(q', u)\| + \|\pi(u, x)\| + \|\pi(y, p')\| \\ &\leq \|\pi(q', u)\| + \|\overline{uv}\| + \|\overline{vx}\| + \|\overline{yw}\| + \|\pi(w, p')\| \\ &\leq \|\pi(q', u)\| + \|\pi(w, p')\| + \|\overline{uv}\| + \|\overline{vy}\| + \|\pi(y, x)\| + \|\overline{yw}\| \\ &= \|\pi(q', u)\| + \|\pi(w, p')\| + (\|\overline{uv}\| + \|\overline{vw}\|) + \|\pi(x, y)\| \end{aligned}$$

In $\Delta_{\pi} uvw$, by Claim 10, we know that $\overline{uv}, \overline{vw}$ are almost parallel. More precisely, $\text{dir}(\overline{uv}) \in \Phi_j$ and $\text{dir}(\overline{vw}) \in \Phi_j$. Since our canonical cones have angle at most $\varepsilon/2$, this means that if we consider the (Euclidean) triangle Δuvw , then the angles at u and w are both less than $\varepsilon/2$. This trivially gives us that

$$\|\overline{uv}\| + \|\overline{vw}\| < \left(1 + \frac{\varepsilon}{2}\right) \|\overline{uw}\| < \left(1 + \frac{\varepsilon}{2}\right) \|\pi(u, w)\|$$

By combining this with our earlier derivation we obtain

$$\begin{aligned} \|\pi(q', x)\| + \|\pi(y, p')\| &< \|\pi(q', u)\| + \|\pi(w, p')\| + \left(1 + \frac{\varepsilon}{2}\right) \|\pi(u, w)\| + \|\pi(x, y)\| \\ &< \left(1 + \frac{\varepsilon}{2}\right) \|\pi(q', p')\| + \|\pi(x, y)\|, \end{aligned}$$

where the last inequality uses that $u, w \in \pi(q', p')$. \triangleleft

With Claim 11 in hand, we are ready to conclude the proof of Subcase II:

$$\begin{aligned} \|\pi(q, p)\| &\leq \|\pi(q, q')\| + \|\pi(q', x)\| + \|\pi(x, \gamma_i)\| + \|\pi(\gamma_i, p)\| && \text{(Triangle Inequality)} \\ &\leq \|\pi(q, q')\| + \|\pi(q', x)\| + \|\Gamma_i\| + \|\pi(\gamma_i, f_{i,j})\| && (x, \gamma_i \in \Gamma_i \text{ and definition of } f_{i,j}) \\ &\leq \|\pi(q, q')\| + \|\pi(q', x)\| + \|\Gamma_i\| + \|\pi(\gamma_i, y)\| + \|\pi(y, f_{i,j})\| && \text{(Triangle Inequality)} \\ &\leq \|\pi(q, q')\| + \|\pi(q', x)\| + 2\|\Gamma_i\| + \|\pi(y, f_{i,j})\| && \text{(since } y, \gamma_i \in \Gamma_i) \\ &\leq \|\pi(q, q')\| + \|\pi(q', x)\| + 2\|\Gamma_i\| + \|\pi(y, p')\| + \|\pi(p', f_{i,j})\| && \text{(Triangle Inequality)} \\ &< \|\pi(q, q')\| + 2\|\Gamma_i\| + \left(1 + \frac{\varepsilon}{2}\right) \|\pi(q', p')\| + \|\pi(x, y)\| + \|\pi(p', f_{i,j})\| && \text{(by Claim 11)} \\ &\leq 3\|\Gamma_i\| + \|\pi(q, q')\| + \left(1 + \frac{\varepsilon}{2}\right) \|\pi(q', p')\| + \|\pi(p', f_{i,j})\| && \text{(since } x, y \in \Gamma_i) \\ &\leq 3\|\Gamma_i\| + \left(1 + \frac{\varepsilon}{2}\right) \|\pi(q, f_{i,j})\| + \|\pi(p', f_{i,j})\| && \text{(since } q', p' \in \pi(q, f_{i,j})) \\ &\leq \frac{\varepsilon}{2} \|\pi(q, p)\| + \left(1 + \frac{\varepsilon}{2}\right) \|\pi(q, f_{i,j})\| && \text{(since } \|\Gamma_i\| \leq \frac{\varepsilon}{6} \|\Gamma\| \leq \frac{\varepsilon}{3} \|\pi(q, p)\|) \end{aligned}$$

and so we have

$$\|\pi(q, f_{i,j})\| > \frac{1 - \varepsilon/2}{1 + \varepsilon/2} \cdot \|\pi(q, p)\| > (1 - \varepsilon) \cdot \|\pi(q, p)\|.$$

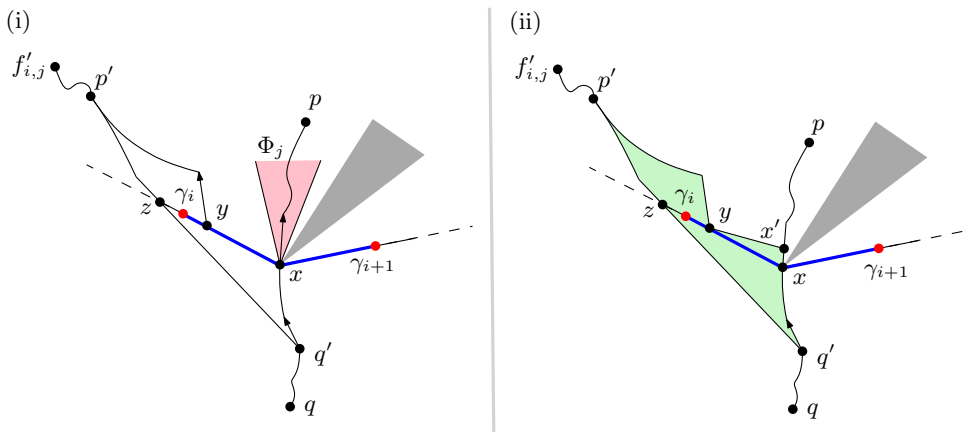
Subcase III: $\pi(q, f_{i,j})$ does not cross Γ_i and $\text{dir}(q', x) \notin \Phi_j$. In this case we also have $\text{dir}(q', p') \notin \Phi_j$, because $\text{first}(\pi(q', p'))$ lies further away from Φ_j than $\text{first}(\pi(q', x))$ does.³ Below we show that this implies $\pi(q, p')$ crosses Γ_i . But then $\pi(q, f_{i,j})$ crosses Γ_i , which contradicts the conditions of Subcase III. Hence, Subcase III cannot occur.

So, assume for a contradiction that $\pi(q, p')$ crosses Γ^* at a point $z \notin \Gamma_i$, as shown schematically in Figure 8. As in Subcase II we may assume that y lies between x and z on Γ^* and thus the angle at y is reflex. Then essentially we can reprove Claim 10 of Subcase II, which will give a contradiction since $\text{dir}(q', p') \notin \Phi_j$. There is a technicality which lies in the case that x coincides with a (reflex) vertex of \mathcal{P} . Then the direction of e_1 , last edge of $\pi(q', x)$, need not be the same as the direction of the first edge of $\pi(x, p)$, and so we may not have $\text{dir}(e_1) \in \Phi_j$. (In Subcase II, because $\text{dir}(q', x) \in \Phi_j$, we know that $\text{dir}(e_1) \in \Phi_j$ even when v coincides with a reflex vertex of \mathcal{P} .) This can be bypassed with the following trick: choose x' in $\text{first}(\pi(x, p))$ very close to x and observe that we can imitate the proof of Claim 10 using $\square_{\pi} q' x' y p'$ instead of $\square_{\pi} q' x y p'$. \blacktriangleleft

2.3 Extension to points outside $\text{RCH}(P)$

We have so far shown how to prove Theorem 1 when the query point q lies inside $\text{RCH}(P)$. When $q \notin \text{RCH}(P)$, it might be that $\pi(q, \text{FN}(q))$ crosses Γ^* , but not Γ . To obtain a valid ε -coreset using the technique of the previous subsections, it thus seems that we should subdivide Γ^* (instead of Γ) into pieces of length $\frac{\varepsilon}{6} \|\Gamma\|$. This is problematic, since Γ^* can be arbitrarily much larger than Γ , and so the size of our coreset may explode. To handle this, we show that there exists a set B consisting of $O(1)$ segments, each of length at most $2\|\Gamma\|$, such that the following holds for any query point $q \notin \text{RCH}(P)$: the shortest path $\pi(q, \text{FN}(q))$ intersects Γ or it intersects a segment in B . We can then split B in $O(1/\varepsilon)$ pieces and then use the ‘‘cone-approach’’ for each piece. The proof can be found in the full version[10].

³ This is true because as we walk back from x to q' along the reflex chain $\pi(q', x)$, the direction turns monotonically. By the time we reach q' , the direction is $\text{dir}(q', x)$, which lies outside Φ_j . When we then turn to $\text{dir}(q', p')$ we turn away from Φ_j even more.



■ **Figure 8** (i) A rough illustration for the case $\text{dir}(q', x) \notin \Phi_j$. Note that in this example x is a reflex vertex of \mathcal{P} . (ii) We can choose x' close to x on $\text{first}(\pi(x, p))$ and repeat the proof of Claim 10 using $\square_{\pi} q' x' y p'$ (green region).

3 An efficient algorithm to construct the coresets

In this section we explain how to construct the coreset C defined in the previous section. We will show that we can do that quite efficiently, namely in time $O\left(\frac{1}{\varepsilon}(n \log(1/\varepsilon) + (n + m) \log(n + m))\right)$.

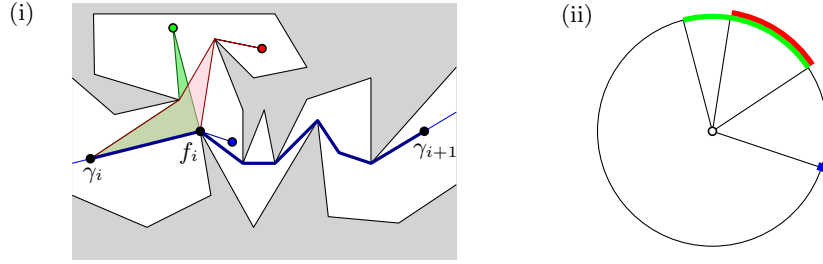
Preprocessing. Our approach crucially uses the diameter of P , so we start by computing Γ . This can be done in $O((m + n) \log(m + n))$ by first computing $\text{RCH}(P)$ in $O(m + n \log(m + n))$ time [23] and then computing its diameter in $O((m + n) \log(m + n))$ time [22]. We then preprocess the subdivision of \mathcal{P} induced by Γ for point location [21], so that we can compute for each point in P whether it lies in $\mathcal{P}_1(\Gamma^*)$ or $\mathcal{P}_2(\Gamma^*)$, in $O(m + n \log m)$ time in total.

We now show how to compute the set C_i of points we put into our coreset C for a given piece Γ_i . We start by preprocessing \mathcal{P} for single-source shortest-path queries with γ_i as the fixed source. This takes $O(m)$ time, after which we can compute the distance from each point in P to γ_i in $O(\log m)$ time [12]. Thus computing all distances takes $O(n \log m)$ time in total. Recall that C_i contains the point furthest away from Γ_{j-1} from each of the following sets.

- The sets $P \cap \mathcal{P}_1(\Gamma^*)$ and $P \cap \mathcal{P}_2(\Gamma^*)$.
- The sets $P \cap E_1$ and $P \cap E_2$, where E_1 and E_2 are the two half-polygons into which \mathcal{P} is split by the intermediate edge e of Γ_i , if such an edge exists.
- The sets $X_i \cap \mathcal{P}_1(\Gamma^*)$ and $X_i \cap \mathcal{P}_2(\Gamma^*)$, where $X_i \subset P$ contains the points that lie in a pocket of Γ_i .
- The sets $F_{i,j} \cap \mathcal{P}_1(\Gamma^*)$ and $L_{i,j} \cap \mathcal{P}_1(\Gamma^*)$, and the set $F_{i,j} \cap \mathcal{P}_2(\Gamma^*)$ and $L_{i,j} \cap \mathcal{P}_2(\Gamma^*)$, where $F_{i,j}$ contains the points reachable from $\text{first}(\Gamma_i)$ in the cone Φ_j , and $L_{i,j}$ contains the points reachable from $\text{last}(\Gamma_i)$ in the cone Φ_j .

The sets $P \cap E_1$, $P \cap E_2$, $X_i \cap \mathcal{P}_1(\Gamma^*)$, $X_i \cap \mathcal{P}_2(\Gamma^*)$ can each be determined by performing a point-location query for each point $p \in P$ in a suitably defined subdivision of complexity m . This can be done in $O((m + n) \log m)$ time in total [21]. Since we precomputed all distances to γ_i , this also gives us the furthest neighbor of γ_i from each of these sets. So far we spent $O((n + m) \log m)$ time in total to compute C_i .

Efficiently determining the furthest neighbors in the sets $F_{i,j} \cap \mathcal{P}_1(\Gamma^*)$, $F_{i,j} \cap \mathcal{P}_2(\Gamma^*)$, $L_{i,j} \cap \mathcal{P}_1(\Gamma^*)$, and $L_{i,j} \cap \mathcal{P}_2(\Gamma^*)$ is slightly more complicated. Next we explain how to handle $F_{i,j} \cap \mathcal{P}_1(\Gamma^*)$; the other sets can be computed analogously.



■ **Figure 9** (i) The shortest paths from three points, shown in green, red, and blue, to γ_i and to f_i . (ii) The directions from which these points can be reached are the directions in between $\text{dir}(\gamma_i, p)$ and $\text{dir}(f_i, p)$. Note that for the blue point these directions are the same.

Consider a point $p \in P$. Recall that we preprocessed \mathcal{P} for single-source shortest-path queries with γ_i as the fixed source. Using this data structure, we can not only find $\|\pi(\gamma_i, p)\|$ in $O(\log m)$ time, but we can also report the first edge of this path in the same amount of time.⁴ Let f_i be the endpoint of $\text{first}(\Gamma_i)$ other than γ_i . We also preprocess \mathcal{P} for single-source shortest-path queries with f_i as the fixed source, so that we can find $\text{first}(\pi(f_i, p))$ in $O(\log n)$ time. After computing these edges, we know all directions ϕ such that there is a point $x \in \text{first}(\Gamma_i)$ with $\text{dir}(x, p) = \phi$, since these are simply the directions in between $\text{dir}(\gamma_i, p)$ and $\text{dir}(f_i, p)$; see Figure 9. This is true since, as x moves along $\text{first}(\Gamma_i)$ from γ_i to f_i , the direction of the first edge changes continuously from $\text{dir}(\cdot, p)$ to $\text{dir}(f_i, p)$. Hence, we can list in $O(\frac{1}{\varepsilon} + \log m)$ time for a given point $p \in P \cap \mathcal{P}_1(\Gamma^*)$ all cone directions for which p is relevant. Thus we can compute all sets $F_{i,j} \cap \mathcal{P}_1(\Gamma^*)$ in $O(n/\varepsilon + n \log m)$ time in total.

We can speed this up using standard range-searching techniques. To this end we build a balanced binary search tree \mathcal{T}_i over the canonical cone directions, seen as sub-intervals of $[0, 360^\circ]$. Because of the way the cones are generated, this can be done in linear time. Each point $p \in P$ induces a set $I(p)$ of one or two sub-intervals of $[0, 360^\circ]$, corresponding to the directions in which p can be reached from $\text{first}(\Gamma_i)$. We assign the distance of p to γ_i as a weight to $I(p)$. Then we want to know, for each cone Φ_j , the maximum weight of any $I(p)$ that such that $\Phi_j \cap I(p) \neq \emptyset$. To compute this we perform a range query with each $I(p)$, selecting a set $N(p)$ of $O(\log(1/\varepsilon))$ nodes in \mathcal{T}_i such that $\Phi_j \cap I(p) \neq \emptyset$ if and only if Φ_j corresponds to a leaf in a subtree rooted at a node in $N(p)$. Then we determine for each node v in \mathcal{T}_i the maximum weight of any $I(p)$ such that $v \in N(p)$, taking $O(n \log(1/\varepsilon))$ time in total. Finally, we traverse \mathcal{T} in a top-down manner to compute for each of its leaves the largest weight associated with any of the nodes on its search path. Thus finding the furthest neighbor from γ_i in $F_{i,j} \cap \mathcal{P}_1(\Gamma^*)$, over all j , takes $O(n \log(1/\varepsilon))$ time in total.

To sum up, for each Γ_i we can compute C_i in time $O(n \log(1/\varepsilon) + (m+n) \log m)$. Since the above computations need to be done for each piece Γ_i and there are $O(1/\varepsilon)$ pieces, the coreset C can be constructed in $O(\frac{1}{\varepsilon}(n \log(1/\varepsilon) + (n+m) \log(n+m)))$ time in total. (Actually, the bound is $O((n+m) \log(n+m) + \frac{1}{\varepsilon}(n \log(1/\varepsilon) + (n+m) \log m))$, but we prefer the slightly shorter expression given above.) This concludes the proof of Theorem 1.

⁴ Indeed, shortest-path queries are answered using a shortest-path tree with the source node as root. It then suffices to know, for each node v in the shortest-path tree, which subtree of the root contains v .

4 Concluding remarks

We proved that any set P of n points in a simple polygon \mathcal{P} of complexity m admits an ε -coreset of size $O(1/\varepsilon^2)$ for furthest-neighbor queries. Thus the size of the coreset is independent of n and m . This immediately gives a data structure for approximate furthest-neighbor queries whose size is independent of n . The size of the data structure will be linear in m , but this is unavoidable: it is easy to see that for $n = 2$ we already have an $\Omega(m)$ lower bound on the storage of any data structure for approximate furthest-neighbor queries.

Another application, which we did not mention so far, is the 1-center problem with z outliers in a simple polygon, for which we can now obtain an ε -coreset of size $O(z/\varepsilon^2)$. To this end we use the technique of Agarwal, Har-Peled, and Yu [2]: compute an ε -coreset C for furthest-neighbor queries on P , remove C from P , compute an ε -coreset for furthest-neighbor queries on the remaining points and add these to the first coreset, and so on. If this is repeated $z + 1$ times, the result is an ε -coreset for the 1-center problem with z outliers.

There are several directions for further research. First of all, we do not expect our bound to be tight: we believe that an ε -coreset of size $O(1/\varepsilon)$, or perhaps even $O(\sqrt{1/\varepsilon})$, should exist. One approach to reduce the size of the coreset may be to show that in our method we can re-use many points, that is, that the $O(1/\varepsilon)$ subsets C_i that we define for each of the $O(1/\varepsilon)$ pieces Γ_i will have (or can be made to have) many points in common. Another challenging open problem is to extend our result to polygons with holes. This seems to require a different approach, as our reduction to points on the diameter fails in this setting.

References

- 1 Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, 2004. doi:10.1145/1008731.1008736.
- 2 Pankaj K. Agarwal, Sariel Har-Peled, and Hai Yu. Robust shape fitting via peeling and grating coresets. In *Proc. 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 182–191, 2006. URL: <http://dl.acm.org/citation.cfm?id=1109557.1109579>.
- 3 Henk Alkema, Mark de Berg, Morteza Monemizadeh, and Leonidas Theocharous. TSP in a Simple Polygon. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *30th Annual European Symposium on Algorithms (ESA 2022)*, volume 244 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:14, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2022.5.
- 4 Boris Aronov, Steven Fortune, and Gordon T. Wilfong. The furthest-site geodesic Voronoi diagram. *Discret. Comput. Geom.*, 9:217–255, 1993. doi:10.1007/BF02189321.
- 5 Franz Aurenhammer, Robert L. Scot Drysdale, and Hannes Krasser. Farthest line segment Voronoi diagrams. *Inf. Process. Lett.*, 100(6):220–225, 2006. doi:10.1016/j.ipl.2006.07.008.
- 6 Franz Aurenhammer, Evanthia Papadopoulou, and Martin Suderland. Piecewise-linear farthest-site Voronoi diagrams. In *32nd International Symposium on Algorithms and Computation (ISAAC)*, volume 212 of *LIPIcs*, pages 30:1–30:11, 2021. doi:10.4230/LIPIcs.ISAAC.2021.30.
- 7 Sergei Bespamyatnikh. Dynamic algorithms for approximate neighbor searching. In *Proceedings of the 8th Canadian Conference on Computational Geometry*, pages 252–257, 1996. URL: http://www.cccg.ca/proceedings/1996/cccg1996_0042.pdf.
- 8 Otfried Cheong, Hazel Everett, Marc Glisse, Joachim Gudmundsson, Samuel Hornus, Sylvain Lazard, Mira Lee, and Hyeon-Suk Na. Farthest-polygon Voronoi diagrams. *Comput. Geom.*, 44(4):234–247, 2011. doi:10.1016/j.comgeo.2010.11.004.
- 9 Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational Geometry: Algorithms and Applications (3rd Edition)*. Springer, 2008. doi:10.1007/978-3-540-77974-2.

- 10 Mark de Berg and Leonidas Theocharous. A coreset for approximate furthest-neighbor queries in a simple polygon. arXiv:2403.04513, 2024. doi:10.48550/arXiv.2403.04513.
- 11 Ashish Goel, Piotr Indyk, and Kasturi R. Varadarajan. Reductions among high dimensional proximity problems. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms*, pages 769–778, 2001. URL: <http://dl.acm.org/citation.cfm?id=365411.365776>.
- 12 Leonidas J. Guibas and John Hershberger. Optimal shortest path queries in a simple polygon. *J. Comput. Syst. Sci.*, 39(2):126–152, 1989. doi:10.1016/0022-0000(89)90041-X.
- 13 Piotr Indyk. Better algorithms for high-dimensional proximity problems via asymmetric embeddings. In *Proc. 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 539–545, 2003. URL: <http://dl.acm.org/citation.cfm?id=644108.644200>.
- 14 Kurt Mehlhorn, Stefan Meiser, and Ronald Rasch. Furthest site abstract Voronoi diagrams. *Int. J. Comput. Geom. Appl.*, 11(6):583–616, 2001. doi:10.1142/S0218195901000663.
- 15 Joseph S. B. Mitchell. Geometric shortest paths and network optimization. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 633–701. North Holland / Elsevier, 2000. doi:10.1016/b978-044482537-7/50016-4.
- 16 Eunjin Oh, Luis Barba, and Hee-Kap Ahn. The geodesic farthest-point Voronoi diagram in a simple polygon. *Algorithmica*, 82(5):1434–1473, 2020. doi:10.1007/s00453-019-00651-z.
- 17 Rasmus Pagh, Francesco Silvestri, Johan Sivertsen, and Matthew Skala. Approximate furthest neighbor with application to annulus query. *Inf. Syst.*, 64:152–162, 2017. doi:10.1016/j.is.2016.07.006.
- 18 Michel Pocchiola and Gert Vegter. The visibility complex. *Int. J. Comput. Geom. Appl.*, 6(3):279–308, 1996. doi:10.1142/S0218195996000204.
- 19 R. Pollack, M. Sharir, and G. Rote. Computing the geodesic center of a simple polygon. *Discrete & Computational Geometry*, 4(6):611–626, December 1989. doi:10.1007/BF02187751.
- 20 Raimund Seidel. On the number of faces in higher-dimensional Voronoi diagrams. In *Proceedings of the Third Annual Symposium on Computational Geometry*, pages 181–185, 1987. doi:10.1145/41958.41977.
- 21 Jack Snoeyink. Point location. In Jacob E. Goodman and Joseph O’Rourke, editors, *Handbook of Discrete and Computational Geometry (3rd edition)*, pages 767–785. Chapman and Hall/CRC, 2017. doi:10.1201/9781315119601.
- 22 Subhash Suri. Computing geodesic furthest neighbors in simple polygons. *Journal of Computer and System Sciences*, 39(2):220–235, 1989. doi:10.1016/0022-0000(89)90045-7.
- 23 Godfried Toussaint. An optimal algorithm for computing the relative convex hull of a set of points in a polygon. In *Proceedings of EURASIP, Signal Processing III: Theories and Applications, Part 2*, pages 853–856, 1986.
- 24 Haitao Wang. An optimal deterministic algorithm for geodesic farthest-point voronoi diagrams in simple polygons. *Discrete & Computational Geometry*, 70(2):426–454, September 2023. doi:10.1007/s00454-022-00424-6.