Convex Hulls in Polygonal Domains

Luis Barba
Department of Computer Science, ETH Zürich, Zürich, Switzerland
luis.barba@inf.ethz.ch

Michael Hoffmann
Department of Computer Science, ETH Zürich, Zürich, Switzerland
hoffmann@inf.ethz.ch
https://orcid.org/0000-0001-5307-7106

Matias Korman[1]
Tohoku University, Sendai, Japan
mati@dais.is.tohoku.ac.jp

Alexander Pilz[2]
Department of Computer Science, ETH Zürich. Zürich, Switzerland
alexander.pilz@inf.ethz.ch
https://orcid.org/0000-0002-6059-1821

Abstract

We study generalizations of convex hulls to polygonal domains with holes. Convexity in Euclidean space is based on the notion of shortest paths, which are straight-line segments. In a polygonal domain, shortest paths are polygonal paths called *geodesics*. One possible generalization of convex hulls is based on the "rubber band" conception of the convex hull boundary as a shortest curve that encloses a given set of sites. However, it is NP-hard to compute such a curve in a general polygonal domain. Hence, we focus on a different, more direct generalization of convexity, where a set X is *geodesically convex* if it contains all geodesics between every pair of points $x, y \in X$. The corresponding *geodesic convex hull* presents a few surprises, and turns out to behave quite differently compared to the classic Euclidean setting or to the geodesic hull inside a simple polygon. We describe a class of geometric objects that suffice to represent geodesic convex hulls of sets of sites, and characterize which such domains are geodesically convex. Using such a representation we present an algorithm to construct the geodesic convex hull of a set of $O(n)$ sites in a polygonal domain with a total of n vertices and h holes in $O(n^3h^{3+\varepsilon})$ time, for any constant $\varepsilon > 0$.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases geometric graph, polygonal domain, geodesic hull, shortest path

Digital Object Identifier 10.4230/LIPIcs.SWAT.2018.8

1 Introduction

Convexity is a fundamental concept in geometry and optimization, and computing the convex hull of a point set in the plane is a classic textbook problem in algorithm design. The convex hull of a set $S \subset \mathbb{R}^2$ is usually defined as the inclusion-minimal convex set that contains S, and showing that this statement is well-defined is a textbook exercise in itself. If S is finite,

[1] Supported in part by MEXT KAKENHI Nos. 17K12635, 15H02665, and 24106007.
[2] Supported by a Schrödinger fellowship of the Austrian Science Fund (FWF): J-3847-N35.

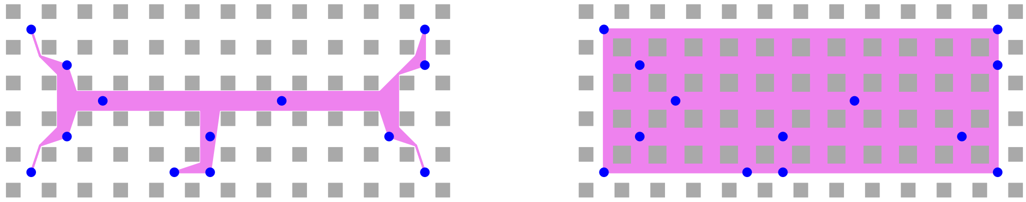


Figure 1 Two possible definitions of "convex hull" in a polygonal domain. The domain is shown in white, obstacles in gray, and sites are shown as blue dots. The left image depicts the relative hull, bounded by a curve of minimum length that separates the set of sites from the boundary. The right image depicts the definition we will use in this paper: an inclusion-minimal subset of the domain that contains all sites and all shortest paths between any two of its points.

the convex hull of S is a convex polygon. The boundary of this polygon describes the shortest path enclosing S, yielding an equivalent definition of the convex hull.

The definition of convexity builds on shortest paths: a set X is convex if for every pair $x, y \in X$ the shortest path between x and y is contained in X. Hence, convexity directly generalizes to any domain that has a notion of a shortest path. In the Euclidean setting, shortest paths are straight-line segments. But there is a variety of other domains that have a sensible notion of a shortest path. Specifically, shortest paths inside a simple polygon have been studied in the computational geometry literature.

A set R is called geodesically convex w.r.t. a polygon P if the shortest path in P between two points in R is also contained in R. Toussaint [20] studied how properties of point sets extend to geodesic environments. He introduced the geodesic convex hull of a set of points (called *sites*) inside a simple polygon P; it is the inclusion-minimal geodesically convex set containing the sites. Among several results, he showed how to compute the geodesic convex hull of k sites in a simple n-gon in $O((n+k)\log(n+k))$ time. Note that the geodesic convex hull properly generalizes the convex hull of a point set S; if we choose P to be, say, a bounding box of S, we obtain the convex hull of S.

A classic metaphor for the convex hull boundary is a "rubber band", describing the continuous transformation of a curve containing the sites to a homotopy-equivalent curve of minimal length. For geodesic convex hulls within a simple polygon P, the boundary ∂P is equivalent to the shortest cycle that separates the sites from ∂P [20]. However, if we consider sites in a polygonal domain with holes, this correspondence does not generalize.

We thus face (at least) two different ways to generalize the concept of a convex hull to general polygonal domains. On the one hand, we have the *(geodesic) convex hull* as an inclusion-minimal geodesically convex set that contains all sites (and may enclose holes). On the other hand, we have a shortest curve that separates the sites from the boundary, also called the *relative hull* of the sites. See Figure 1 for illustrations.

Both generalizations are interesting in their own right. The former definition is much more directly tied to the notions of convexity and shortest paths. Therefore this is how we propose to generalize the concept *convex hull* to general polygonal domains. The latter definition using relative hulls turns out rather unwieldy. For a set S of general sites inside a polygonal domain, a relative hull is not necessarily unique and NP-hard to compute. This follows from a slight modification of a result by Eades and Rappaport [12], who show that it is NP-hard to find the shortest curve separating two point sets. (A reduction from the rectilinear Steiner tree problem is also straightforward.)

Related work. Relative hulls have been studied in general polygonal domains, but only for a set of connected sites. Given a set of disjoint simple polygons with n vertices overall, de Berg [10] showed how to compute the shortest curve that separates one of these polygons from the others in $O(n \log n)$ time. Effectively, the algorithm computes the shortest cycle within a polygonal domain that separates a polygon P from the boundary. The proof directly generalizes to the case where P is an arbitrary outerplane graph. In a similar fashion, Mitchell et al. [18] compute the relative hull of paths in polynomial time.

In addition to Toussaint's generalization of the diameter, center, and median to the geodesic setting [20], separators [11], ham-sandwich cuts [7], spanning trees, Hamiltonian cycles and perfect matchings [6] have been generalized to point sites in simple polygons. Any concept defined on the order type of a point set allows for a generalization [2]. In general polygonal domains, the complexity of these problems increases substantially. Many problems become NP-hard, and where polynomial algorithms are known, the known bounds are nowhere near to what is known for simple polygons. For example, the diameter and center of a simple polygon can be computed in linear time [1, 14]. However, for a general domain, the best known algorithms use $O(n^{7.73})$ [3] and $O(n^{11} \log n)$ [4, 21] time, respectively.

Computing shortest paths in polygonal domains has been an active area of research (cf. [17]). While a single shortest path can be computed in $O(n \log n)$ time [15], data structures that support two-point shortest path queries in logarithmic time require a significant storage overhead. The state of the art data structure, allowing $O(\log n)$ query time, uses $O(n^{11})$ space and preprocessing time [9]. For points on the boundary of the domain, Bae and Okamoto [5] presented a data structure with logarithmic query time using $O(n^{5+\varepsilon})$ space and preprocessing time. A variant of their result is used as a subroutine in our algorithm.

Generalizations of convex hulls of point sets have also been considered in other settings. For example, Lubiw et al. [16] consider convex hulls in 2-dimensional globally non-positively curved polyhedral complexes. Such spaces have a unique shortest path between any two points. They pose as an open problem the study of convexity in domains where more than one shortest path between two points may exist. Our work is a step in this direction.

Results. We consider the inclusion-minimum geodesically convex set that contains a given set of sites in a polygonal domain. It is the first study of this natural generalization of convex hulls. Not even domains with a single hole have been considered so far (see also [16], where the problem is mentioned). It turns out that the problem of computing the geodesic convex hull within a polygonal domain is significantly more complex than within a simple polygon. Within a simple polygon, the structure of the geodesic convex hull only depends on the order type of the sites and the vertices, i.e., the orientation of point triples. In general polygonal domains, the homotopy of the shortest path between two sites depends on actual distances between sites and vertices. In particular, all direct attempts to discretize the problem failed. The examples given in Section 2 illustrate the differences to classic convexity and demonstrate how naive attempts to compute the geodesic convex hull fail.

As a main result, we characterize geodesically convex sets. To this end, we define a class of geometric objects, called *cactus domains*, and show that this class contains all geodesic convex hulls of finite sets of sites inside polygonal domains. More specifically, we use two concepts (called divisibility and tightness), and show that they are sufficient and necessary for a cactus domain to be geodesically convex. We provide algorithms to efficiently test both properties, resulting in a polynomial-time algorithm to compute geodesic convex hulls.

▶ **Theorem 1.** *Let P be a polygonal domain with n vertices and h holes, and let $S \subset P$ be a set of $O(n)$ sites. The geodesic convex hull of S in P can be computed in $O(n^3 h^{3+\varepsilon})$ time.*

While the running time of our algorithm might look high at first sight, it must be compared with algorithms and data structures that encode all geodesic paths in polygonal domains. In this direction, one must consider the state-of-the-art structure developed by Chiang and Mitchell [9] that uses $O(n^{11})$ space and preprocessing time; or the structure of Bae and Okamoto [5] using $O(n^{5+\varepsilon})$ space and preprocessing time for paths connecting points on the boundary. While no lower bounds are known, it is clear that the complexity of these problems is high and still far from being understood.

To improve upon the running time stated in Theorem 1, more structural insights would be required. As a first step in this direction, one could ask if a simpler algorithm can be designed to test whether a point lies in the geodesic convex hull of a set of sites in a polygonal domain.

2 Preliminaries

Polygonal domains. A *simple polygon* is a compact subset of $\mathbb{R}^2$ that is bounded by a simple closed curve formed by a finite number of line segments. For a simple polygon P denote by $\mathsf{V}(P)$ the set of its *vertices*, by $\mathsf{int}(P)$ the *interior* of P, and by ∂P its *boundary*. A *polygonal domain* P is defined by a finite collection $(P_0, P_1, \ldots, P_h)$ of $h+1$ simple polygons with the following properties: (1) $P_i \subset \mathsf{int}(P_0)$, for each $i > 0$, and (2) $P_i \cap P_j = \emptyset$, for all $i, j > 0$ with $i \neq j$. We say that P_0 and ∂P_0 are the *outer polygon* and *outer boundary* of P, respectively. The *boundary* of P is $\partial P = \bigcup_{i=0}^{h} \partial P_i$, the *interior* of P is $\mathsf{int}(P) = \mathsf{int}(P_0) \setminus \bigcup_{i=1}^{h} P_i$, the *vertices* of P are $\mathsf{V}(P) = \bigcup_{i=0}^{h} \mathsf{V}(P_i)$, and collectively $P = \mathsf{int}(P) \cup \partial P$. The polygons $P_1, \ldots, P_h$ are also referred to as *holes* of P. We also use the notation $P = (P_0, P_1, \ldots, P_h)$ to indicate that P is defined by the polygons $P_0, P_1, \ldots, P_h$ (although in principle we regard P as a subset of the plane rather than a tuple of polygons).

Geodesic convex hulls. In the following consider a polygonal domain P with n vertices. For two points $x, y \in P$ denote by $\Pi_P(x, y)$ the set of *geodesics* between x and y in P. That is, every element of $\Pi_P(x, y)$ is a curve from x to y that is contained in P and corresponds to a shortest path between x and y (among all curves between x and y in P). A set $K \subseteq P$ is *geodesically convex* (in P) if, for every $x, y \in K$, all geodesics in $\Pi_P(x, y)$ are contained in K. For $S \subseteq P$, the *geodesic convex hull*, or simply G*-hull* of S in P, is the (inclusion) minimum geodesically convex set $\mathsf{GH}_P(S) \subseteq P$ that contains S. In this paper, we study the case in which S consists of a finite set of $O(n)$ points (called *sites*).

One way to conceive the G-hull of S is to start with $C_0 = S$ and iteratively add more points as follows. In the i-th step, for every pair of points $x, y \in C_{i-1}$ (possibly infinitely many) take all geodesics in $\Pi_P(x, y)$ and add them to the new set C_i. Continue until $C_i = C_{i-1}$ at the end of some step. Note that this procedure as described is not an algorithm because (i) the number of pairs/geodesics to consider is not finite in general and (ii) it is not clear whether the procedure terminates after a finite number of steps.

Visibility graphs and shortest path maps. Every geodesic in $\Pi_P(x, y)$, for $x, y \in S$, forms a path in the *visibility graph* $\mathrm{Vis}_P(S)$ of S with respect to P. This graph is defined on the vertex set $V = S \cup \mathsf{V}(P)$ and two vertices $x, y \in V$ are *visible* and connected by an edge in $\mathrm{Vis}_P(S)$ if the relative open line segment $\overline{xy} \setminus \{x, y\}$ is contained in $P \setminus V$. For given P and S, the graph $\mathrm{Vis}_P(S)$ can be computed in $O(|V|^2)$ time and space [22].

For a point $s \in P$, the *shortest path map* (SPM) for s is the subdivision of P into *cells* to which the geodesic from s passes through the same sequence of vertices of P. There are $O(n)$

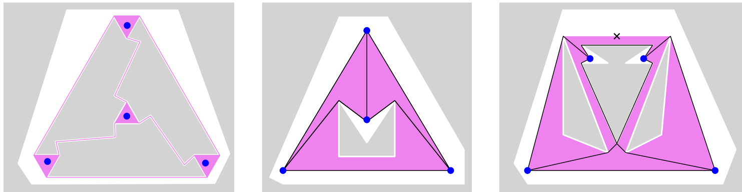
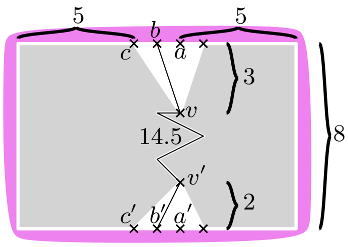


Figure 2 No site appears in the boundary (left). In the middle figure there is no way of partitioning the four sites so that the convex hulls of the two sets intersect. To the right, the top point (cross) belongs to the G-hull of the four sites, but it is not included in the G-hull of any three sites. The geodesics between pairs of sites are shown in black.



$$
\begin{aligned}
&d(v,v') = 14.5\\
&d(a,a') = 18 < d(a,v) + d(v,v') + d(v',a') = 19.5\\
&d(b,b') = \sqrt{5} + \sqrt{10} + 14.5 \approx 19.79\\
&\qquad < 20 = d(b,a) + d(a,a') + d(a',b')\\
&d(a,c') = 20 < d(a,v) + d(v,v') + d(v',c') \approx 20.33\\
&d(a',c) = 20 < d(a',v') + d(v,v') + d(v,c) \approx 20.1\\
&d(a,b') = 19 < d(a,v) + d(v,v') + d(v',b') \approx 19.73\\
&d(a',b) = 19 < d(a',v') + d(v,v') + d(v,b) \approx 19.66
\end{aligned}
$$

Figure 3 A partial drawing of a domain (violet), in which eight points have been highlighted. The geodesic between b and b' passes through v and v', while the geodesics between pairs a and a' or c and c' do not pass through v or v'. The function $d(\cdot,\cdot)$ denotes the geodesic distance.

such cells, and the boundaries between these cells are formed by curves of constant algebraic degree. Hershberger and Suri [15] provide an $O(n \log n)$ time algorithm to construct the SPM for a given point s. Given the SPM for s, we can compute the geodesic distance from s to any query point $p \in P$ in $O(\log n)$ time using point location. In the same time, we can also get the first and last vertex (other than s and p, if any) of some path in $\Pi_P(s,p)$.

Remarkable properties of g-hulls. Figure 2 depicts a polygon P with all sites in the interior of $\mathsf{GH}_P(S)$, as well as an example where an analogue of Radon's Theorem does not hold, i.e., there is no partition of S into two non-empty sets $S_1 \cup S_2$ such that $\mathsf{GH}_P(S_1) \cap \mathsf{GH}_P(S_2) \neq \emptyset$. Similarly, Figure 2 (right) depicts an example where the natural extension of Carathéodory's Theorem does not hold: there exists a point in $\mathsf{GH}_P(S)$ that does not belong to the G-hull of any three sites of S. An example where the actual distance between points influences the structure of the G-hull is given in the (partial) instance depicted in Figure 3. Moving the points slightly without changing the order type can have large influence on the structure of the G-hull.

3 Cactus domains and general properties

Even in a single simple polygon, the G-hull of two segments on its boundary forms a so-called funnel [13], which, in general, is not simple. It is therefore natural to study a slightly more general class of polygons to be able to describe G-hulls. A frequently used relaxation is referred to as a weakly simple polygon, which, intuitively speaking, allows the curve that describes the boundary to touch but not properly cross itself. However, to make this intuition formally precise is surprisingly cumbersome [8]. For describing G-hulls, a much more restricted class of polygons is sufficient, which we will define in the next paragraphs. (We refrain from using the term "weakly simple polygon" to emphasize this difference.)

In a *plane drawing* or *embedding* of a connected graph, the vertices are represented by pairwise distinct points, edges are represented by Jordan arcs connecting their endpoints, and no two edges intersect except at a common endpoint. In a *straight-line* drawing, all Jordan arcs representing edges are (straight) line segments. A *cactus* is a connected graph in which every edge belongs to at most one simple cycle. Cacti are *outerplanar*, that is, they can be embedded in the plane so that all vertices are incident to one particular face (which is usually the outer face).

A *cactusgon* is a domain K that is represented by an outerplane straight-line drawing $\varphi(G)$ of a cactus G: The interior $\mathsf{int}(K)$ is formed by the union of all (open) bounded faces in $\varphi(G)$, the boundary ∂K is formed by the union of all edges in $\varphi(G)$, and collectively $K = \mathsf{int}(K) \cup \partial K$. We obtain a combinatorial representation of ∂K as the unique circular sequence of edges and vertices as they appear along the boundary of the outer face of $\varphi(G)$ in counterclockwise order. A closed curve that traverses ∂K of some cactusgon K in this fashion is a *cactus curve*. Note that K is a compact subset of $\mathbb{R}^2$ and that every simple polygon is a cactusgon whose associated graph G is a simple cycle. Consider a face of a cactus that is incident to all vertices (which may or may not be unbounded); we call this open subset of the plane a *cactus face*. The boundary of a cactus face is defined analogously to the one of a cactusgon (but note that the boundary is not part of the cactus face).

▶ **Definition 2.** A *cactus domain* or, for short, *C-domain* K is a planar region bounded by an *outer polygon* $\mathcal{C}$ and $t \geq 0$ *inner voids* $K_1, \ldots, K_t$, where (1) $\mathcal{C}$ is a cactusgon, (2) $K_i \subseteq \mathcal{C}$ is a bounded cactus face, for $1 \leq i \leq t$, and $K_1, \ldots, K_t$ are pairwise disjoint. The *outer void* K_0 of K is is an unbounded cactus face (or, equivalently, the outer face of $\mathcal{C}$). The interior of K is $\mathsf{int}(K) = \mathbb{R}^2 \setminus \bigcup_{i=0}^{t}(K_i \cup \partial K_i)$, the boundary of K is $\partial K = \bigcup_{i=0}^{t} \partial K_i$, the vertices of K are $\mathsf{V}(K) = \bigcup_{i=0}^{t} \mathsf{V}(K_i)$, and collectively $K = \mathsf{int}(K) \cup \partial K$.

Note that the above definition slightly abuses notation since $V(\cdot)$ was only defined for polygons. Along the paper we do a similar abuse for structures defined for polygons (such as shortest path map and visibility graph) and apply them to cactusgons. The extension of these concepts (and the algorithims) are straighforward. Thus, for simplicity we omit them.

Observe that $\partial\mathcal{C} = \partial K_0$ and so $\mathsf{V}(\mathcal{C}) = \mathsf{V}(K_0)$. Again, we write $K = (\mathcal{C}, K_1, \ldots, K_t)$ as a shorthand. As usual, $\mathsf{int}(K)$ is an open subset of $\mathbb{R}^2$ and K is a compact subset of $\mathbb{R}^2$ with $\mathbb{R}^2 \setminus K = \bigcup_{i=0}^{t} \mathsf{int}(K_i)$. Observe that the cycles of K may share edges or points of their boundary. As an extreme example, if $\mathcal{C}$ is simple, then we may even have $t = 1$ and one large hole $K_1 = \mathcal{C}$ (in this case, K is just a one-dimensional polygonal cycle). While the theorem below is not hard to prove in a stand-alone way, it will follow from our algorithmic construction of G-hulls, as our algorithm produces a C-domain that we prove to coincide with the G-hull of S.

▶ **Theorem 3.** *Given a polygonal domain P with n vertices in total and a set $S \subset P$ of $O(n)$ sites, the geodesic convex hull $\mathsf{GH}_P(S)$ of S in P is a cactus domain whose vertices are from $S \cup \mathsf{V}(P)$ and whose edges are edges of the visibility graph $\mathrm{Vis}_P(S)$. In particular, the boundary of $\mathsf{GH}_P(S)$ can be described as a plane straight-line graph on $O(n)$ vertices.*

4 Characterization of geodesically convex sets

The aim of this section is to give a characterization of C-domains that are geodesically convex in a polygonal domain $P = (P_0, \ldots, P_h)$. Consider a C-domain $K = (\mathcal{C}, K_1, \ldots, K_t)$. If K is not geodesically convex, then there exist two points $p, q \in K$ and a geodesic $\pi \in \Pi_P(p, q)$ such that $\pi \not\subset K$. For simplicity, assume that $\pi \cap K = \{p, q\}$. (That is, the geodesic only

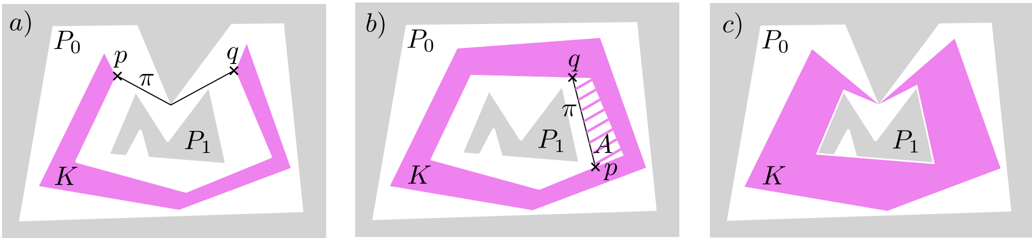


Figure 4 *a)* A C-domain K that is divisible by π. *b)* An indivisible C-domain K that is not tight. *c)* An indivisible and tight C-domain K.

touches K at the endpoints. This can be achieved by restricting π.) As $K_0, \ldots, K_t$ are pairwise interior-disjoint, p and q lie in the same component of ∂K, say ∂K_i. Therefore, π splits the void K_i into two parts A and B; refer to Figure 4 for illustrations.

In the case in which one of the two parts, say A, contains no hole of P and also not its outer face (Figure 4b), we use a local operation to enlarge K following the rubber band metaphor. We show in Lemma 4 that all of A is in $\mathsf{GH}_P(K)$. A C-domain without such a geodesic is called tight.

The other possible situation is that both A and B contain at least one hole or the outer face of P. In this case we have a path that is topologically different from all paths in K, and we say that K is divisible by π (Figure 4a). If no such path exists, then K is indivisible.

Clearly, any geodesically convex C-domain must be indivisible and tight. In the remainder of this section, we prove in form of a characterization that the reverse implication holds as well, i.e., a C-domain K is geodesically convex if and only if it is indivisible and tight.

Tightness of cactus domains. For $1 \leq i \leq t$, let $\mathsf{V}_K(i) = \bigcup_{P_j \subseteq K_i} \mathsf{V}(P_j)$ be the vertices of all holes P_j of P for which P_j is contained in the void K_i. Observe that, in general, ∂K_i may contain vertices of holes of P not contained in K_i. Thus, $\mathsf{V}_K(i)$ may be different from the set of vertices of P contained in K_i. As P_0 is not contained in any inner void of K, for the outer void, we let $\mathsf{V}_K(0) = \mathsf{V}(P_0) \cup \bigcup_{P_j \subseteq K_0} \mathsf{V}(P_j)$. In particular, $\mathsf{V}_K(0) \neq \emptyset$.

A curve γ *separates* two compact subsets $A, B \subset \mathbb{R}^2$ if every curve that connects a point in A with a point in B intersects γ. Given a void K_i of K with $\mathsf{V}_K(i) \neq \emptyset$, we define the *reduction* $\varrho(K_i)$ as the minimum length curve in P that separates $\mathsf{V}_K(i)$ from $\mathsf{int}(K)$ (possibly $\varrho(K_i) = \partial K_i$). We can think of $\varrho(K_i)$ as being obtained by continuously tightening a curve tracing ∂K_i as much as possible while maintaining separation between $\mathsf{V}_K(i)$ and K.

Algorithmically, an inner void K_i, $i \geq 1$, can usually be treated as a simple polygon. It follows from Toussaint's algorithm [20] that $\varrho(K_i)$ is a (non-simple, in general) closed walk in $\mathrm{Vis}_{K_i}(\mathsf{V}_K(i))$; in fact, $\varrho(K_i)$ is a cactus curve. Similarly, for the outer void K_0 the outside domain is formed by the outer void and a collection of simple polygons (P_0 and possibly some holes in the exterior of K). The algorithm of de Berg [10] asserts that $\varrho(K_0)$ is a cactus curve in this case as well.

For an inner void K_i, $i \geq 1$, the boundary ∂K_i encloses $\varrho(K_i)$. For the outer void K_0, the curve $\varrho(K_0)$ encloses ∂K_0; see Figure 5. Regardless, ∂K_i and $\varrho(K_i)$ form an annulus (possibly with no interior point). We say that any point in this annulus lies *between* ∂K_i and $\varrho(K_i)$. Given a C-domain K, a void K_i of K is *tight* if $\mathsf{V}_K(i) \neq \emptyset$ and $\partial K_i = \varrho(K_i)$. We say that K is *tight* if K_i is tight for each $0 \leq i \leq t$. If K is tight, then for each $1 \leq i \leq t$, the void K_i contains at least one hole of P; otherwise, $\mathsf{V}_K(i)$ would be empty.

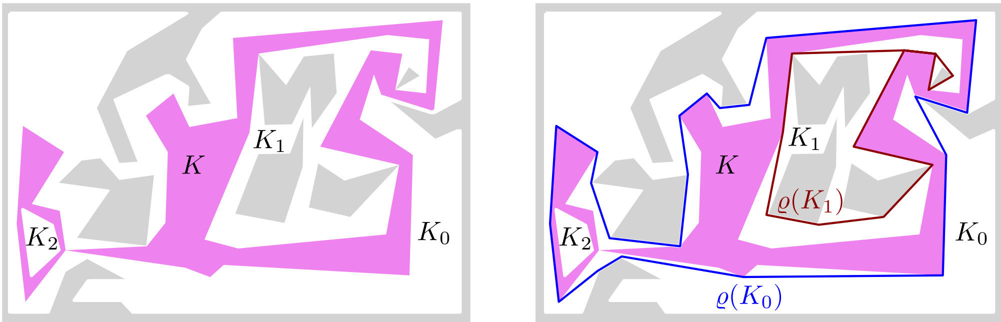


Figure 5 Left: A C-domain K with two inner voids. Right: The reductions of the voids K_0 and K_1 are depicted. The curve $\varrho(K_1)$ is enclosed by ∂K_1, while ∂K_0 is enclosed by $\varrho(K_0)$ (curves shown in solid red and blue, respectively). Every point between ∂K_i and $\varrho(K_i)$ belongs to $\mathsf{GH}_P(K)$. Notice that, since K_2 has no hole, we have $K_2 \subseteq \mathsf{GH}_P(K)$.

▶ **Lemma 4.** *Let P be a polygonal domain, $K = (\mathcal{C}, K_1, \ldots, K_t)$ a C-domain in P, and K_i a void of K, for $0 \leq i \leq t$.* (1) *If $\mathsf{V}_K(i) \neq \emptyset$, then each point that lies between ∂K_i and $\varrho(K_i)$ belongs to $\mathsf{GH}_P(K)$.* (2) *If $\mathsf{V}_K(i) = \emptyset$, then $K_i \subseteq \mathsf{GH}_P(K)$.*

Characterization of geodesically convex cactus domains. Using the above lemma, we are ready to give sufficient and necessary conditions for a C-domain to be geodesically convex. It remains to formally define divisibility. Given a void K_i, we say that two points $p, q \in \partial K_i$ *separate* K_i if a geodesic in $\Pi_P(p, q)$, called *separating geodesic*, splits K_i into two connected components, each containing either at least one hole of P or the outer face of P; see Figure 4a. We say that K_i is *divisible* if some pair of points on ∂K_i separates K_i. Analogously, K_i is *indivisible* if no pair of points on ∂K_i separates K_i. A C-domain is *divisible* if at least one of its voids is divisible; otherwise, it is *indivisible*.

▶ **Theorem 5.** *A C-domain K is geodesically convex if and only if it is indivisible and tight.*

Theorem 5 is the main structural result our algorithm relies on. To algorithmically test divisibility of a C-domain, we use the following lemma.

▶ **Lemma 6.** *If a C-domain K is divisible, then there exists a geodesic π separating a void K_i with the following three properties. (i) The intersection of π with ∂K consists only of its endpoints. (ii) It contains a vertex of P in its relative interior or both of its endpoints are vertices of K_i. (iii) It consists of at least one segment that intersects the interior of K_i.*

5 Computing the geodesic convex hull

In this section, we present an algorithm that, given a polygonal domain P and a set S of sites, computes $\mathsf{GH}_P(S)$. To simplify the presentation and the analysis, we assume that $|S| = O(n)$, where n is the number of vertices of P. But the exact dependency on $|S|$ can be easily derived from our proofs.

Our algorithm is founded upon the characterization of Theorem 5. We first start with the C-domain formed by the union of all geodesics going from an arbitrary site of S to all other sites. As a next step, we make use of Lemma 4 on the resulting C-domain to obtain a new tight C-domain that we test for divisibility. If this tight C-domain is divisible, our procedure reports a geodesic that separates it, which we add to the C-domain. The addition of this separating geodesic generates a new C-domain that is not necessarily tight. We repeat the procedure iteratively until we obtain a C-domain that is both tight and indivisible. Then by Theorem 5 this domain is $\mathsf{GH}_P(S)$.

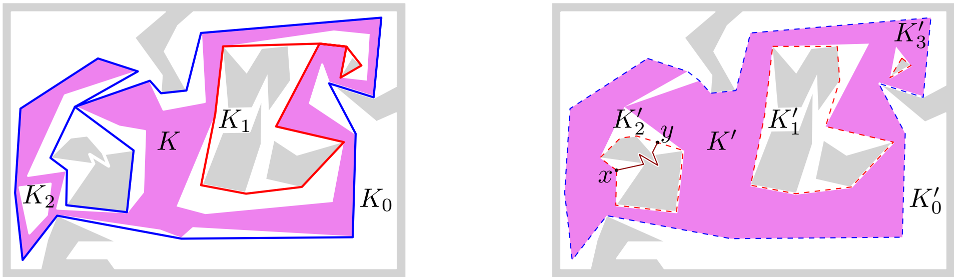


Figure 6 Left: A C-domain K and the reduction of each of its voids. Right: The new C-domain K' obtained from K after applying steps (1) and (2) of the tightening process. In the left figure we have two inner voids. The reduction of K_0 and K_1 are shown in solid blue and red curves, respectively (K_2 need not be reduced because it has no holes). The right figure shows the resulting C-domain K'. This domain is not tight, so step (2) needs to be applied a second time (the region to be added is shown dashed). Note that the indivisible void K'_2 may become divisible after the tightening process (due to the geodesic between points x and y).

Computing tightenings of cactus domains. We introduce the *tightening process* of a C-domain $K = (\mathcal{C}, K_1, \ldots, K_t)$. Intuitively speaking, we want to enlarge K as little as possible until it is tight. The result is another C-domain K', which we call the *tightening* of K. In order to do so, we proceed as follows: (1) for each $1 \leq i \leq t$ such that $\mathsf{V}_K(i) = \emptyset$, we add each point in this void to the tightening of K (effectively removing this void from K), and (2) for each $0 \leq i \leq t$ such that $\mathsf{V}_K(i) \neq \emptyset$, compute the reduction of K_i and add the space in the annulus between ∂K_i and $\varrho(K_i)$ to the tightening of K. Recall that the reduction $\varrho(K_i)$ of a void K_i of K need not be a simple curve. Therefore, to obtain a valid C-domain, we consider each bounded component of $\mathbb{R}^2 \setminus \varrho(K_i)$ (which are cactus faces) and add them as new voids replacing K_i. In particular, the resulting C-domain may have more voids than K (and they need not be tight, see Figure 6). Thus, we apply again step (2) iteratively until we obtain a C-domain in which the boundary of each void coincides with its reduction. Since every newly created void needs to contain a hole of P, we obtain this C-domain after at most h iterations. Since in the resulting domain the boundary of each void coincides with its reduction, we obtain a tight C-domain K', the tightening of K.

▶ **Lemma 7.** *Given a C-domain K with $O(n)$ vertices, we can compute the tightening of K in $O(hn \log n)$ time. Moreover, the tightening of K is a C-domain whose edges belong to* $\mathrm{Vis}_P(\mathsf{V}(K))$.

Testing divisibility of cactus domains. In this section we provide a deterministic algorithm to determine if a C-domain is divisible. This property is considerably harder to test than tightness. In fact, this test is the main bottleneck of our algorithm and the main algorithmic challenge of this paper.

Let $K = (\mathcal{C}, K_1, \ldots, K_t)$ be a tight C-domain. To test the divisibility of K, we test each void K_i separately. Using Lemma 6, it is sufficient to determine whether there is a separating geodesic containing a vertex of P in its relative interior or that is a segment between two vertices of K_i that see each other. The latter can be easily tested using the visibility graph of P. For testing the former, we modify an algorithm by Bae and Okamoto [5]: this $O(n^{5+\varepsilon})$-time algorithm takes a polygonal domain on n vertices, and encodes all geodesics between pairs of points on its boundary as the lower envelope of a collection of constant-degree distance functions. While their algorithm serves to construct a data structure for shortest-path queries among points on the boundary of a polygonal domain, we are able to translate its main ideas to test divisibility. Additionally, several new observations allow us to replace a factor of $O(n^{2+\varepsilon})$ for a factor of $O(h^{2+\varepsilon})$ in the running time. The remaining part of this section describes our algorithm in detail.

As a preprocessing step, compute the SPM from every vertex of P in overall $O(n^2 \log n)$ time [15]. Then, for each edge e of K_i, split e at each point of intersection with the boundary of a cell in the SPM of some vertex of P. In this way, we obtain the *spm-subdivison* of e into *spm-segments*. The *spm-subdivision* of ∂K_i is the union of the spm-subdivision of its edges.

Let $\text{SPM}(p)$ be the SPM for a point p. We claim that if, for some vertex v of P and cell c of $\text{SPM}(v)$, c intersects ∂K_i in three or more connected components, then there is a segment t contained in this cell connecting two points of ∂K_i through the interior of K_i. Moreover, t must be a separating geodesic, as otherwise t would split K_i into two components, one of which would not contain a vertex of $\mathsf{V}_K(i)$. However, since t can be used as a shortcut to reduce the length of ∂K_i while separating $\mathsf{V}_K(i)$ from K, we obtain a contradiction with the tightness of K_i, which proves our claim. Thus, if a cell of the SPM of some vertex of P intersects ∂K_i in three or more connected components, then K_i is divisible.

Therefore, to compute the spm-subdivision, we first compute the intersection points of the SPM of each vertex with ∂K_i, and then sort all these intersection points along the boundary of K_i to obtain the spm-subdivision of ∂K_i. If at some point during this process we find a cell of a SPM that intersects ∂K_i in more than two connected components, then the algorithm finishes and reports the separating geodesic contained in this cell. Thus, we assume from now on that no cell of an SPM intersects ∂K_i in more than two connected components, i.e., each cell of an SPM contributes to $O(1)$ spm-segments to the spm-subdivision. Because we consider the SPM of the n vertices of P, each with $O(n)$ cells, the spm-subdivision of K consists of $O(n^2)$ spm-segments, and the total running time of our preprocessing step is bounded by $O(n^2 \log n)$.

An important property of the spm-subdivision is that for a spm-segment s and a point $x \in s$, the set of vertices of P that are visible from x remains unchanged as x moves along s. Thus, we let V_s be the set of vertices of P visible from s. For a pair of spm-segments s and s', each geodesic with at least two segments from a point in s to a point in s' *starts* with a vertex v in V_s (i.e., v is the first vertex visited by this path after leaving s). Moreover, because s' is contained in a single cell of $\text{SPM}(v)$, a geodesic from v to any point in s' must have the same combinatorial structure. Let v^* be the last vertex visited in the path from v to any point of s' (note that we may have $v = v^*$). Then, any path from a point $x \in s$ to a point $y \in s'$ can be parametrized by the distance function $f_v(x, y) = d(x, v) + d(v, v^*) + d(v^*, y)$. Because $d(v, v^*)$ is a known constant, f_v is a constant degree algebraic function from $s \times s'$ to $\mathbb{R}$. We could then compute the minimization diagram of the set $F_{s,s'} = \{f_v(x, y) : v \in V_s\}$, i.e., the lower envelope of these distance functions over all different starting vertices. This diagram has the following property: the algebraic surface patch of f_v appears in this lower envelope if and only if there is a geodesic from a point $x \in s$ to a point $y \in s'$ that passes through v. We now look for a vertex v that lies in the interior of K_i and f_v appears in the lower envelope. If this happens, there is a separating geodesic connecting s with s' starting at v (and thus we conclude that K_i is divisible); note that by Lemma 6 this is sufficient to determine divisibility. This gives us an algorithm to decide divisibility whose running time is dominated by the computation of $O(n^4)$ minimization diagrams, one for each pair of spm-segments. We will improve this later but first, we look in more detail at the starting vertices of the geodesics we need to consider. The following observation leads to our main improvement when compared to the algorithm of Bae and Okamoto [5].

▶ **Lemma 8.** *Given an spm-segment s and a hole H of P, there are at most two starting vertices in H among all geodesics going from a point in s to a point in ∂K_i. Moreover, they are the counterclockwise- and clockwise-most vertices in $V_s \cap V(H)$, when sorted radially around any point in s.*

Therefore, at most two geodesics from s to ∂K_i can start at different vertices of $V_s \cap V(H)$. That is, each hole can contribute to at most two start vertices, hence only a total of $O(h)$ starting vertices must be considered.

By Lemma 8, we can let $V_s^* \subseteq V_s$ denote the set of $O(h)$ starting vertices of paths going from s to ∂K_i. Moreover, we can compute V_s^* in $O(n)$ time by computing the maximum and minimum element, among the vertices of each hole of P, in the radial order around an arbitrary point of s. Because we need to consider only $O(h)$ vertices in V_s^*, we notice that there are many divisions among spm-segments that do not correspond to the boundary of a cell in the SPM of a vertex in V_s^*. Thus, we could modify our spm-subdivision with respect to s and consider only the breaking points induced by the SPM of a vertex in V_s^*. Because each SPM has complexity $O(n)$ and since $|V_s^*| = O(h)$, this induces at most $O(nh)$ divisions. In this way, we obtain a partition of ∂K_i into $O(nh)$ *s-segments*, each being a collection of consecutive spm-segments. The idea of using this subdivision is that, to compute a minimization diagram of distance functions between s and an s-segment, we need to consider only $O(h)$ functions defined by the vertices in V_s^*.

▶ **Theorem 9.** *We can determine if a tight C-domain K of $O(n)$ vertices in a polygonal domain $P = (P_0, \dots, P_h)$ of n vertices is divisible in $O(n^3 h^{2+\varepsilon})$ time.*

Proof. Let s be a spm-segment. Note that, when going from one s-segment to a neighboring one, the SPM cell of at most one vertex in V_s^* can change. Intuitively, this means that the distance functions we need to consider have "little" variation among neighboring s-segments. We formalize this intuition as follows. Group h consecutive s-segments lying on the same edge of ∂K_i and take their union to produce an *s-block* g. We claim that $O(h)$ distance functions need to be considered to compute the minimization diagram encoding all geodesics from s to any s-block g. To show this, for each $v \in V_s^*$, let τ_v be the number of cells of $\text{SPM}(v)$ that intersect s-block g. Let σ be a cell of $\text{SPM}(v)$ that intersects g. Notice that there is exactly one ending vertex v^* in any geodesic from v to $\sigma \cap g$. Thus, we can define an *s-g-function* $f_{v,\sigma} : s \times (\sigma \cap g) \to \mathbb{R}$ such that $f_{v,\sigma}(x, y) = d(x, v) + d(v, v^*) + d(v^*, y)$. Note that there are exactly τ_v s-g-functions defined for each vertex v of V_s^*. Because g consists of h s-segments, we know that g can be intersected by at most $O(h)$ cells among the SPMs of the vertices in V_s^*. Therefore, $\sum_{v \in V_s^*} \tau_v = O(h)$, i.e., there are in total $O(h)$ s-g-functions defined for all vertices of V_s^*. Moreover, any geodesic from s to g needs to start with a vertex of V_s^* and hence, it is considered in one of these functions. Consequently, the minimization diagram of the s-g-functions encodes the distance of all geodesics going from s to g. Note that this minimization diagram can be computed in $O(h^{2+\varepsilon})$ time [19]. After computing it, we can check within the same time whether there is a geodesic between s and g that goes through the interior of K_i by going through all elements of this lower envelope.

By grouping all $O(nh)$ s-segments into consecutive s-blocks of at most h spm-segments, each contained in a single edge of ∂K_i, we obtain $O(n)$ s-blocks in total along ∂K_i. Therefore, we need to compute $O(n)$ minimization diagrams for a given spm-segment s, one for each s-block, each in $O(h^{2+\varepsilon})$ time. Repeating this over all $O(n^2)$ spm-segments gives a total running time of $O(n^3 h^{2+\varepsilon})$. ◀

▶ **Theorem 1.** *Let P be a polygonal domain with n vertices and h holes, and let $S \subset P$ be a set of $O(n)$ sites. The geodesic convex hull of S in P can be computed in $O(n^3 h^{3+\varepsilon})$ time.*

Proof. Let s be a site of S. For each $s' \in S \setminus \{s\}$, choose an arbitrary path in $\Pi_P(s, s')$. Let K^0 be the plane connected graph obtained by taking the union of all chosen paths. Notice that K^0 is a connected C-domain that contains all sites of S. Moreover, because K^0 is plane (as no two geodesics from s can cross), K^0 consists of $O(n)$ vertices and edges and $K^0 \subseteq \mathsf{GH}_P(S)$ (as it consists of geodesics between points of S). We describe now a recursive procedure that incrementally constructs the G-hull of S starting from K^0.

Given a C-domain K^r for some even number r, we construct K^{r+1} as the tightening of K^r using Lemma 7 in $O(hn \log n)$ time. Since P has h holes, K^r is a tight C-domain with at most h voids whose vertices and edges are contained in $\mathrm{Vis}_P(S)$. Thus, because K^{r+1} is plane, it has complexity $O(n)$. We then use Theorem 9 to test whether K^{r+1} is divisible or not, which takes $O(n^3 h^{2+\varepsilon})$ time. If K^{r+1} is indivisible, then as it is also tight, Theorem 5 implies that K^{r+1} is geodesically convex. Thus, as $S \subset K^{r+1}$ and since $\mathsf{GH}_P(S)$ is the smallest geodesically convex set that contains S, we get that $\mathsf{GH}_P(S) \subseteq K^{r+1}$. Moreover, because all points in K^{r+1} belong to $\mathsf{GH}_P(S)$ by Lemma 4, we know that $K^{r+1} \subseteq \mathsf{GH}_P(S)$. Therefore, if K^{r+1} is indivisible, then $K^{r+1} = \mathsf{GH}_P(S)$ and we are done.

Otherwise K^{r+1} is divisible and we have found a separating geodesic, i.e., there is some void of K^{r+1} and two points x and y such that the path $\pi_P(x, y)$ separates K^{r+1}. In this case, we add the path $\pi_P(x, y)$ to K^{r+1} and obtain a new C-domain $K^{r+2} \subset \mathsf{GH}_P(S)$ that is not necessarily tight. Because $r + 2$ is even, we can repeat this procedure recursively until finding a tight indivisible C-domain. One may think that one test for divisibility suffices, i.e., that this does not need to be repeated every time that a tightening is computed. However, the tightening of an indivisible C-domain may be divisible; see Figure 6.

Note that in each round, if the tight C-domain K^{r+1} is divisible, then we find a new separating geodesic that separates two holes of P that were previously in the same void of K^{r+1}. In particular, we create a new void with at least one hole. Since we can have at most h such voids, the above procedure will iterate at most h times and must end with a tight indivisible domain that coincides with $\mathsf{GH}_P(S)$.

The running time is dominated by the divisibility test given by Theorem 9 which has to be executed at most h times. Thus, the total running time becomes $O(n^3 h^{3+\varepsilon})$ as claimed. ◀

References

1. Hee-Kap Ahn, Luis Barba, Prosenjit Bose, Jean-Lou De Carufel, Matias Korman, and Eunjin Oh. A linear-time algorithm for the geodesic center of a simple polygon. *Discrete Comput. Geom.*, 56(4):836–859, 2016.
2. Oswin Aichholzer, Matias Korman, Alexander Pilz, and Birgit Vogtenhuber. Geodesic order types. *Algorithmica*, 70(1):112–128, 2014. doi:10.1007/s00453-013-9818-8.
3. Sang Won Bae, Matias Korman, and Yoshio Okamoto. The geodesic diameter of polygonal domains. *Discrete Comput. Geom.*, 50(2):306–329, 2013.
4. Sang Won Bae, Matias Korman, and Yoshio Okamoto. Computing the geodesic centers of a polygonal domain. In *Proc. 26th Canadian Conf. on Computational Geometry*, 2014.
5. Sang Won Bae and Yoshio Okamoto. Querying two boundary points for shortest paths in a polygonal domain. *Comput. Geom.*, 45(7):284–293, 2012. doi:10.1016/j.comgeo.2012.01.012.
6. Ahmad Biniaz, Prosenjit Bose, Anil Maheshwari, and Michiel H. M. Smid. Plane geodesic spanning trees, hamiltonian cycles, and perfect matchings in a simple polygon. *Comput. Geom.*, 57:27–39, 2016. doi:10.1016/j.comgeo.2016.05.004.
7. Prosenjit Bose, Erik D. Demaine, Ferran Hurtado, John Iacono, Stefan Langerman, and Pat Morin. Geodesic ham-sandwich cuts. *Discrete & Computational Geometry*, 37(3):325–339, 2007. doi:10.1007/s00454-006-1287-2.

8 Hsien-Chih Chang, Jeff Erickson, and Chao Xu. Detecting weakly simple polygons. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1655–1670. SIAM, 2015. doi:10.1137/1.9781611973730.110.

9 Yi-Jen Chiang and Joseph S. B. Mitchell. Two-point Euclidean shortest path queries in the plane. In *Proc. 10th ACM-SIAM Symposium on Discrete Algorithms*, 1999.

10 Mark de Berg. Translating polygons with applications to hidden surface removal. In John R. Gilbert and Rolf G. Karlsson, editors, *SWAT 90, 2nd Scandinavian Workshop on Algorithm Theory, Bergen, Norway, July 11-14, 1990, Proceedings*, volume 447 of *Lecture Notes in Computer Science*, pages 60–70. Springer, 1990. doi:10.1007/3-540-52846-6_78.

11 Erik D. Demaine, Jeff Erickson, Ferran Hurtado, John Iacono, Stefan Langerman, Henk Meijer, Mark H. Overmars, and Sue Whitesides. Separating point sets in polygonal environments. *Int. J. Comput. Geometry Appl.*, 15(4):403–420, 2005. doi:10.1142/S0218195905001762.

12 Peter Eades and David Rappaport. The complexity of computing minimum separating polygons. *Pattern Recognition Letters*, 14(9):715–718, 1993. doi:10.1016/0167-8655(93)90140-9.

13 Leonidas J. Guibas and John Hershberger. Optimal shortest path queries in a simple polygon. *J. Comput. Syst. Sci.*, 39(2):126–152, 1989.

14 John Hershberger and Subhash Suri. Matrix searching with the shortest-path metric. *SIAM J. Comput.*, 26(6):1612–1634, 1997.

15 John Hershberger and Subhash Suri. An optimal algorithm for euclidean shortest paths in the plane. *SIAM J. Comput.*, 28(6):2215–2256, 1999. doi:10.1137/S0097539795289604.

16 Anna Lubiw, Daniela Maftuleac, and Megan Owen. Shortest paths and convex hulls in 2d complexes with non-positive curvature. *CoRR*, abs/1603.00847, 2016. arXiv:1603.00847.

17 Joseph S. B. Mitchell. Shortest paths and networks. In *Handbook of Discrete and Computational Geometry, Second Edition*. Chapman and Hall/CRC, 2004.

18 Joseph S. B. Mitchell, Günter Rote, and Gerhard J. Woeginger. Minimum-link paths among obstacles in the plan. *Algorithmica*, 8(5&6):431–459, 1992. doi:10.1007/BF01758855.

19 Micha Sharir. Almost tight upper bounds for lower envelopes in higher dimensions. *Discrete Comput. Geom.*, 12(3):327–345, 1994.

20 Godfried T. Toussaint. Computing geodesic properties inside a simple polygon. *Revue D'Intelligence Artificielle*, 3(2):9–42, 1989.

21 Haitao Wang. On the geodesic centers of polygonal domains. In *Proc. 24rd Annual European Symposium on Algorithms*, volume 57 of *LIPIcs*, pages 77:1–77:17, 2016.

22 Emo Welzl. Constructing the visibility graph for n-line segments in $o(n^2)$ time. *Inf. Process. Lett.*, 20(4):167–171, 1985. doi:10.1016/0020-0190(85)90044-4.