Computing the L_1 Geodesic Diameter and Center of a Polygonal Domain

Sang Won Bae[1], Matias Korman[2], Joseph S. B. Mitchell[3], Yoshio Okamoto[4], Valentin Polishchuk[5], and Haitao Wang[6]

1 **Kyonggi University, Suwon, South Korea**
`swbae@kgu.ac.kr`
2 **Tohoku University, Sendai, Japan**
`mati@dais.is.tohoku.ac.jp`
3 **Stony Brook University, New York, USA**
`jsbm@ams.stonybrook.edu`
4 **The University of Electro-Communications, Tokyo, Japan**
`okamotoy@uec.ac.jp`
5 **Linköping University, Linköping, Sweden**
`valentin.polishchuk@liu.se`
6 **Utah State University, Utah, USA**
`haitao.wang@usu.edu`

Abstract

For a polygonal domain with h holes and a total of n vertices, we present algorithms that compute the L_1 geodesic diameter in $O(n^2+h^4)$ time and the L_1 geodesic center in $O((n^4+n^2h^4)\alpha(n))$ time, where $\alpha(\cdot)$ denotes the inverse Ackermann function. No algorithms were known for these problems before. For the Euclidean counterpart, the best algorithms compute the geodesic diameter in $O(n^{7.73})$ or $O(n^7(h+\log n))$ time, and compute the geodesic center in $O(n^{12+\epsilon})$ time. Therefore, our algorithms are much faster than the algorithms for the Euclidean problems. Our algorithms are based on several interesting observations on L_1 shortest paths in polygonal domains.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, I.1.2 Algorithms, I.3.5 Computational Geometry and Object Modeling

Keywords and phrases geodesic diameter, geodesic center, shortest paths, polygonal domains, L_1 metric

Digital Object Identifier 10.4230/LIPIcs.STACS.2016.14

1 Introduction

A *polygonal domain* $\mathcal{P}$ is a closed and connected polygonal region in the plane $\mathbb{R}^2$, with $h \geq 0$ holes (i.e., simple polygons). Let n be the total number of vertices of $\mathcal{P}$. Regarding the boundary of $\mathcal{P}$ as obstacles, we consider shortest obstacle-avoiding paths lying in $\mathcal{P}$ between any two points $p, q \in \mathcal{P}$. Their *geodesic distance* $d(p,q)$ is the length of a shortest path between p and q in $\mathcal{P}$. The *geodesic diameter* (or simply *diameter*) of $\mathcal{P}$ is the maximum geodesic distance over all pairs of points $p, q \in \mathcal{P}$, i.e., $\max_{p\in\mathcal{P}} \max_{q\in\mathcal{P}} d(p,q)$. Closely related to the diameter is the min-max quantity $\min_{p\in\mathcal{P}} \max_{q\in\mathcal{P}} d(p,q)$, in which a point p^* that minimizes $\max_{q\in\mathcal{P}} d(p^*,q)$ is called a *geodesic center* (or simply *center*) of $\mathcal{P}$. Each of the above quantities is called *Euclidean* or L_1 depending on which of the Euclidean or L_1 metric is adopted to measure the length of paths.

For simple polygons (i.e., $h = 0$), the Euclidean diameter and center have been studied since the 1980s [2, 8, 23]. Hershberger and Suri [16] gave a linear-time algorithm for computing

33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016).
Editors: Nicolas Ollinger and Heribert Vollmer; Article No. 14; pp. 14:1–14:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the diameter. Pollack, Sharir, and Rote [21] gave an $O(n \log n)$ time algorithm for computing the geodesic center; recently, Ahn et al. [1] solved the problem in $O(n)$ time. For the general case (i.e., $h > 0$), the Euclidean diameter problem was solved in $O(n^{7.73})$ or $O(n^7(h + \log n))$ time [4], and the Euclidean center problem was solved in $O(n^{12+\epsilon})$ time for any $\epsilon > 0$ [5].

For the L_1 versions, the diameter and the center of simple polygons can be computed in linear time [6, 22]. In this paper, we present the first algorithms that compute the diameter and center of a polygonal domain $\mathcal{P}$ (as defined above) in $O(n^2 + h^4)$ and $O((n^4 + n^2h^4)\alpha(n))$ time, respectively, where $\alpha(\cdot)$ is the inverse Ackermann function. Comparing with the algorithms for the same problems under the Euclidean metric, our algorithms are much more efficient, especially when h is significantly smaller than n.

As discussed in [4], a main difficulty of polygonal domains seemingly arises from the fact that there can be several topologically different shortest paths between two points, which is not the case for simple polygons. Bae, Korman, and Okamoto [4] observed that the Euclidean diameter can be realized by two interior points of a polygonal domain, in which case the two points have at least five distinct shortest paths. This difficulty makes their algorithm suffer a fairly large running time. Similar issues also arise in the L_1 metric, where a diameter may also be realized by two interior points (this can be seen by easily extending the examples in [4]). Further, under the L_1 metric, it seems that at least eight topologically different shortest paths are needed to pin the solution; thus, even if we manage to adapt all techniques used in [4] to the L_1 metric, this would result in an algorithm whose running time is significantly larger than $O(n^{7.73})$.

We take a different approach from [4]. We first construct an $O(n^2)$-sized cell decomposition of $\mathcal{P}$ such that the L_1 geodesic distance function restricted in any pair of two cells can be explicitly described in $O(1)$ complexity. Consequently, the L_1 diameter and center can be obtained by exploring these cell-restricted pieces of the geodesic distance. This leads to simple algorithms that compute the diameter in $O(n^4)$ time and the center in $O(n^6\alpha(n))$ time. With the help of an "extended corridor structure" of $\mathcal{P}$ [9, 10, 11, 12], we reduce the $O(n^2)$ complexity of our decomposition to another "coarser" decomposition of $O(n + h^2)$ complexity; with another crucial observation (Lemma 7), one may compute the diameter in $O(n^3 + h^4)$ time by using our techniques for the above $O(n^4)$ time algorithm. One of our main contributions is an additional series of observations (Lemmas 9 to 18) that allow us to further reduce the running time to $O(n^2 + h^4)$. These observations along with the decomposition may also have other applications. The idea for computing the center is similar.

We are motivated to study the L_1 versions of the diameter and center problems in polygonal (even non-rectilinear) domains for several reasons. First, the L_1 metric is natural and well studied in optimization and routing problems, as it models actual costs in rectilinear road networks and certain robotics/VLSI applications. Indeed, the L_1 diameter and center problems in the simpler setting of simply connected domains have been studied [6, 22]. Second, the L_1 metric approximates the Euclidean metric. Further, improved understanding of algorithmic results in one metric can assist in understanding in other metrics.

1.1 Preliminaries

For any subset $A \subset \mathbb{R}^2$, denote by ∂A the boundary of A. Denote by $\overline{pq}$ the line segment with endpoints p and q. For any path $\pi \in \mathbb{R}^2$, let $|\pi|$ be the L_1 *length* of π. A path is *xy-monotone* (or *monotone* for short) if every vertical or horizontal line intersects it in at most one connected component.

► **Fact 1.** *For any monotone path π between two points $p, q \in \mathbb{R}^2$, $|\pi| = |\overline{pq}|$ holds.*

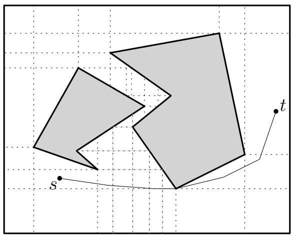

Figure 1 The cell decomposition $\mathcal{D}$ of $\mathcal{P}$, and a shortest path from s to t.

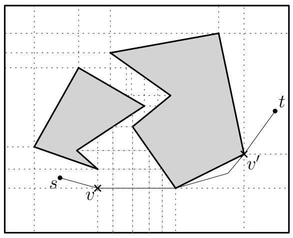

Figure 2 Illustrating Lemma 1: a shortest path through vertices $v \in V_\sigma$ and $v' \in V_{\sigma'}$.

We view the boundary $\partial\mathcal{P}$ of $\mathcal{P}$ as a series of *obstacles* so that no path in $\mathcal{P}$ is allowed to cross $\partial\mathcal{P}$. Throughout the paper, unless otherwise stated, a shortest path always refers to an L_1 shortest path and the distance/length of a path (e.g., $d(p, q)$) always refers to its L_1 distance/length. The diameter/center always refers to the L_1 geodesic diameter/center.

▶ **Fact 2** ([14, 15]). *In any simple polygon P, there is a unique Euclidean shortest path π between any two points in P. The path π is also an L_1 shortest path in P.*

The rest of the paper is organized as follows. In Section 2, we introduce our cell decomposition of $\mathcal{P}$ and exploit it to have preliminary algorithms for computing the diameter and center of $\mathcal{P}$. The algorithms will be improved later in Section 4, based on the extended corridor structure and new observations discussed in Section 3.

Due to the space limit, most lemma and theorem proofs are omitted but can be found in the full version of the paper [3].

2 The Cell Decomposition and Preliminary Algorithms

We first build the *horizontal trapezoidal map* by extending a horizontal line from each vertex of $\mathcal{P}$ until each end of the line hits $\partial\mathcal{P}$. Next, we compute the *vertical trapezoidal map* by extending a vertical line from each vertex of $\mathcal{P}$ *and* each of the ends of the above extended lines. We then overlay the two trapezoidal maps, resulting in a *cell decomposition* $\mathcal{D}$ of $\mathcal{P}$ (e.g., see Fig. 1). The above extended horizontal or vertical line segments are called the *diagonals* of $\mathcal{D}$. Note that $\mathcal{D}$ has $O(n)$ diagonals and $O(n^2)$ cells. Each cell σ of $\mathcal{D}$ appears as a trapezoid or a triangle; let V_σ be the set of vertices of $\mathcal{D}$ that are incident to σ (note that $|V_\sigma| \le 4$). We let $\mathcal{D}$ also denote the set of all the cells of the decomposition.

Each cell of $\mathcal{D}$ is an intersection between a trapezoid of the horizontal trapezoidal map and another one of the vertical trapezoidal map. Two cells of $\mathcal{D}$ are *aligned* if they are contained in the same trapezoid of the horizontal or vertical trapezoidal map, and *unaligned* otherwise. Lemma 1 is crucial for computing both the diameter and the center of $\mathcal{P}$.

▶ **Lemma 1.** *Let σ, σ' be any two cells of $\mathcal{D}$. For any point $s \in \sigma$ and any point $t \in \sigma'$, if σ and σ' are aligned, then $d(s, t) = |\overline{st}|$; otherwise, there exists an L_1 shortest path between s and t that passes through two vertices $v \in V_\sigma$ and $v' \in V_{\sigma'}$ (e.g., see Fig. 2).*

2.1 Computing the Geodesic Diameter

The general idea is to consider every pair of cells of $\mathcal{D}$ separately. For each pair of such cells $\sigma, \sigma' \in \mathcal{D}$, we compute the maximum geodesic distance between σ and σ', that is, $\max_{s\in\sigma, t\in\sigma'} d(s, t)$, called the (σ, σ')-*constrained diameter*. Since $\mathcal{D}$ is a decomposition of $\mathcal{P}$, the diameter of $\mathcal{P}$ is equal to the maximum value of the constrained diameters over all pairs of cells of $\mathcal{D}$. We handle two cases depending on whether σ and σ' are aligned.

If σ and σ' are aligned, by Lemma 1, for any $s \in \sigma$ and $t \in \sigma'$, we have $d(s,t) = |\overline{st}|$, i.e, the L_1 distance of $\overline{st}$. Since the L_1 distance function is convex, the (σ, σ')-constrained diameter is always realized by some pair (v, v') of two vertices with $v \in V_\sigma$ and $v' \in V'_\sigma$. We are thus done by checking at most 16 pairs of vertices, in $O(1)$ time.

In the following, we assume that σ and σ' are unaligned. Consider any point $s \in \sigma$ and any point $t \in \sigma'$. For any vertex $v \in V_\sigma$ and any vertex $v' \in V_{\sigma'}$, consider the path from s to t obtained by concatenating $\overline{sv}$, a shortest path from v to v', and $\overline{v't}$, and let $d_{vv'}(s,t)$ be its length. Lemma 1 ensures that $d(s,t) = \min_{v \in V_\sigma, v' \in V_{\sigma'}} d_{vv'}(s,t)$. Since $d_{vv'}(s,t) = |\overline{sv}| + |\overline{v't}| + d(v,v')$ and $d(v,v')$ is constant over all $(s,t) \in \sigma \times \sigma'$, the function $d_{vv'}$ is linear on $\sigma \times \sigma'$. Thus, it is easy to compute the (σ, σ')-constrained diameter once we know the value of $d(v,v')$ for every pair (v,v') of vertices.

▶ **Lemma 2.** *For any two cells $\sigma, \sigma' \in \mathcal{D}$, the (σ, σ')-constrained diameter can be computed in constant time, provided that $d(v,v')$ for every pair (v,v') with $v \in V_\sigma$ and $v' \in V_{\sigma'}$ has been computed.*

For each vertex v of $\mathcal{D}$, an easy way can compute $d(v,v')$ for all other vertices v' of $\mathcal{D}$ in $O(n^2 \log n)$ time, by first computing the shortest path map $SPM(v)$ [19, 20] in $O(n \log n)$ time and then computing $d(v,v')$ for all $v' \in \mathcal{D}$ in $O(n^2 \log n)$ time. We instead have a faster algorithm in Lemma 3, due to that all vertices on every diagonal of $\mathcal{D}$ are sorted.

▶ **Lemma 3.** *For each vertex v of $\mathcal{D}$, we can evaluate $d(v,v')$ for all vertices v' of $\mathcal{D}$ in $O(n^2)$ time.*

Thus, after $O(n^4)$-time preprocessing, for any two cells $\sigma, \sigma' \in \mathcal{D}$, the (σ, σ')-constrained diameter can be computed in $O(1)$ time by Lemma 2. Since $\mathcal{D}$ has $O(n^2)$ cells, it suffices to handle at most $O(n^4)$ pairs of cells, resulting in $O(n^4)$ candidates for the diameter, and the maximum is the diameter.

▶ **Theorem 4.** *The L_1 geodesic diameter of $\mathcal{P}$ can be computed in $O(n^4)$ time.*

2.2 Computing the Geodesic Center

For any point $q \in \mathcal{P}$, we define $R(q)$ to be the maximum geodesic distance between q and any point in $\mathcal{P}$, i.e., $R(q) := \max_{p \in \mathcal{P}} d(p,q)$. A center q^* of $\mathcal{P}$ is defined to be a point with $R(q^*) = \min_{q \in \mathcal{P}} R(q)$. Our approach is again based on the decomposition $\mathcal{D}$: for each cell $\sigma \in \mathcal{D}$, we want to find a point $q \in \sigma$ that minimizes the maximum geodesic distance $d(p,q)$ over all $p \in \mathcal{P}$. We call such a point $q \in \sigma$ a *σ-constrained center*. Thus, if q' is a σ-constrained center, then we have $R(q') = \min_{q \in \sigma} R(q)$. Clearly, the center q^* of $\mathcal{P}$ must be a σ-constrained center for some $\sigma \in \mathcal{D}$. Our algorithm thus finds a σ-constrained center for every $\sigma \in \mathcal{D}$, which at last results in $O(n^2)$ candidates for a center of $\mathcal{P}$.

Consider any cell $\sigma \in \mathcal{D}$. To compute a σ-constrained center, we investigate the function R restricted to σ and exploit Lemma 1 again. To utilize Lemma 1, we define $R_{\sigma'}(q) := \max_{p \in \sigma'} d(p,q)$ for any $\sigma' \in \mathcal{D}$. For any $q \in \sigma$, $R(q) = \max_{\sigma' \in \mathcal{D}} R_{\sigma'}(q)$, that is, R is the upper envelope of all the $R_{\sigma'}$ on the domain σ. Our algorithm explicitly computes the functions $R_{\sigma'}$ for all $\sigma' \in \mathcal{D}$ and computes the upper envelope $\mathcal{U}$ of the graphs of the $R_{\sigma'}$. Then, a σ-constrained center corresponds to a lowest point on $\mathcal{U}$.

▶ **Lemma 5.** *The function $R_{\sigma'}$ is piecewise linear on σ and has $O(1)$ complexity.*

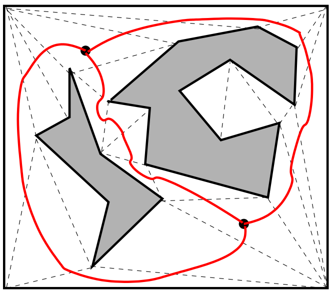

Figure 3 A triangulation $\mathcal{T}$ of $\mathcal{P}$ and the 3-regular graph obtained from the dual graph of $\mathcal{T}$ whose nodes and edges are depicted by black dots and red solid curves.

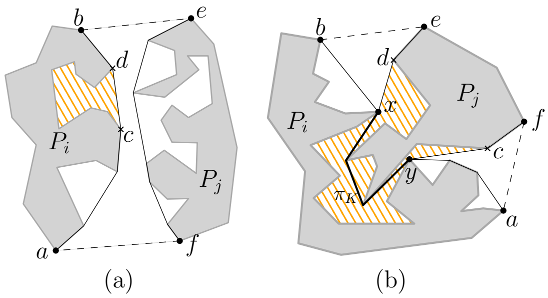

Figure 4 Hourglasses H_K in corridors K. (a) H_K is open. Five bays can be seen. A bay with gate $\overline{cd}$ is shown as the shaded region. (b) H_K is closed. There are three bays and a canal, and the shaded region depicts the canal with two gates $\overline{dx}$ and $\overline{cy}$.

To compute a σ-constrained center, we first handle every cell $\sigma' \in \mathcal{D}$ to compute the graph of $R_{\sigma'}$ and thus gather its linear patches. Let Γ be the family of those linear patches for all $\sigma' \in \mathcal{D}$. We then compute the upper envelope of Γ and find a lowest point on the upper envelope, which corresponds to a σ-constrained center. Since $|\Gamma| = O(n^2)$ by Lemma 5, the upper envelope can be computed in $O(n^4\alpha(n))$ time by executing the algorithm by Edelsbrunner et al. [13], where $\alpha(\cdot)$ denotes the inverse Ackermann function.

▶ **Theorem 6.** *An L_1 geodesic center of $\mathcal{P}$ can be computed in $O(n^6\alpha(n))$ time.*

3 Exploiting the Extended Corridor Structure

In this section, we briefly review the extended corridor structure of $\mathcal{P}$ and present new observations, which will be crucial for our improved algorithms in Section 4. The corridor structure has been used for solving shortest path problems [9, 17, 18]. Later some new concepts such as "bays," "canals," and the "ocean" were introduced [10, 11], referred to as the "extended corridor structure."

3.1 The Extended Corridor Structure

Let $\mathcal{T}$ denote an arbitrary triangulation of $\mathcal{P}$ (e.g., see Figure 3). We can obtain $\mathcal{T}$ in $O(n \log n)$ time or $O(n + h \log^{1+\epsilon} h)$ time for any $\epsilon > 0$ [7]. Based on the dual graph of $\mathcal{T}$, one can obtain a planar 3-regular graph, possibly with loops and multi-edges, by repeatedly removing all degree-one nodes and then contracting all degree-two nodes. The resulting 3-regular graph has $O(h)$ faces, nodes, and edges [18]. Each node of the graph corresponds to a triangle in $\mathcal{T}$, called a *junction triangle*. The removal of all junction triangles from $\mathcal{P}$ results in $O(h)$ components, called *corridors*, each of which corresponds to an edge of the graph. See Figure 3. Refer to [18] for more details.

Let $P_1, \ldots, P_h$ be the h holes of $\mathcal{P}$ and P_0 be the outer polygon of $\mathcal{P}$. For simplicity, a hole may also refer to the unbounded region outside P_0 hereafter. The boundary ∂K of a corridor K consists of two diagonals of $\mathcal{T}$ and two paths along the boundary of holes P_i and P_j, respectively (it is possible that P_i and P_j are the same hole, in which case one may consider P_i and P_j as the above two paths respectively). Let $a, b \in P_i$ and $e, f \in P_j$ be the endpoints of the two paths, respectively, such that $\overline{be}$ and $\overline{fa}$ are diagonals of $\mathcal{T}$, each of which bounds a junction triangle. See Figure 4. Let π_{ab} (resp., π_{ef}) denote the *Euclidean* shortest path from a to b (resp., e to f) inside K. The region H_K bounded by π_{ab}, π_{ef}, $\overline{be}$, and $\overline{fa}$ is called an *hourglass*, which is either *open* if $\pi_{ab} \cap \pi_{ef} = \emptyset$, or *closed*, otherwise. If H_K is open, then both π_{ab} and π_{ef} are convex chains and are called the *sides* of H_K; otherwise, H_K consists of two "funnels" and a path $\pi_K = \pi_{ab} \cap \pi_{ef}$ joining the two apices of the two funnels, called the *corridor path* of K. The two funnel apices (e.g., x and y in Figure 4(b)) connected by π_K are called the *corridor path terminals.* Note that each funnel comprises two convex chains.

We consider the region of K minus the interior of H_K, which consists of a number of simple polygons facing (i.e., sharing an edge with) one or both of P_i and P_j. We call each of these simple polygons a *bay* if it is facing a single hole, or a *canal* if it is facing both holes. Each bay is bounded by a portion of the boundary of a hole and a segment $\overline{cd}$ between two obstacle vertices c, d that are consecutive along a side of H_K. We call the segment $\overline{cd}$ the *gate* of the bay. (See Figure 4(a).) On the other hand, there exists a unique canal for each corridor K only when H_K is closed and the two holes P_i and P_j both bound the canal. The canal in K in this case completely contains the corridor path π_K. A canal has two *gates* $\overline{xd}$ and $\overline{yc}$ that are two segments facing the two funnels, respectively, where x, y are the corridor path terminals and d, c are vertices of the funnels. (See Figure 4(b).)

Let $\mathcal{M} \subseteq \mathcal{P}$ be the union of all junction triangles, open hourglasses, and funnels. We call $\mathcal{M}$ the *ocean.* Its boundary $\partial\mathcal{M}$ consists of $O(h)$ convex vertices and $O(h)$ reflex chains each of which is a side of an open hourglass or of a funnel. Note that each bay or canal is a simple polygon and $\mathcal{P} \setminus \mathcal{M}$ consists of all bays and canals of $\mathcal{P}$.

For convenience of discussion, we define each bay/canal in such a way that they do not contain their gates and hence their gates are contained in $\mathcal{M}$; therefore, each point of $\mathcal{P}$ is either in a bay/canal or in $\mathcal{M}$, but not in both. The following lemma is one of our key observations for our improved algorithms in Section 4.

▶ **Lemma 7.** *Let $s \in \mathcal{P}$ be any point and A be a bay or canal of $\mathcal{P}$. Then, for any $t \in A$, there exists $t' \in \partial A$ such that $d(s,t) \le d(s,t')$. Equivalently, $\max_{t\in A} d(s,t) = \max_{t\in\partial A} d(s,t)$.*

3.2 Shortest Paths in the Ocean $\mathcal{M}$

We now discuss shortest paths in $\mathcal{M}$. Recall that corridor paths are contained in canals, but their terminals are on $\partial\mathcal{M}$. By using the corridor paths and $\mathcal{M}$, finding an L_1 or Euclidean shortest path between two points s and t in $\mathcal{M}$ can be reduced to the convex case since $\partial\mathcal{M}$ consists of $O(h)$ convex chains. For example, suppose both s and t are in $\mathcal{M}$. Then, there must be a shortest s-t path π that lies in the union of $\mathcal{M}$ and all corridor paths [9, 11, 18].

Consider any two points s and t in $\mathcal{M}$. A shortest s-t path $\pi(s,t)$ in $\mathcal{P}$ is a shortest path in $\mathcal{M}$ that possibly contains some corridor paths. Intuitively, one may view corridor paths as "shortcuts" among the components of the space $\mathcal{M}$. As in [18], since $\partial\mathcal{M}$ consists of $O(h)$ convex vertices and $O(h)$ reflex chains, the complementary region $\mathcal{P}' \setminus \mathcal{M}$ (where $\mathcal{P}'$ refers to the union of $\mathcal{P}$ and all its holes) can be partitioned into a set $\mathcal{B}$ of $O(h)$ convex objects with a total of $O(n)$ vertices (e.g., by extending an angle-bisecting segment inward from each convex vertex [18]). If we view the objects in $\mathcal{B}$ as obstacles, then π is a shortest path avoiding all obstacles of $\mathcal{B}$ but possibly containing some corridor paths. Note that our algorithms can work on $\mathcal{P}$ and $\mathcal{M}$ directly without using $\mathcal{B}$; but for ease of exposition, we will discuss our algorithm with the help of $\mathcal{B}$.

Each convex obstacle P of $\mathcal{B}$ has at most four *extreme vertices*: the topmost, bottommost, leftmost, and rightmost vertices, and there may be some corridor path terminals on the boundary of P. We connect the extreme vertices and the corridor path terminals on ∂P

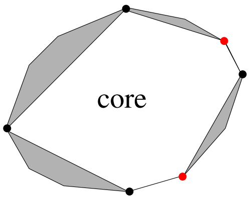

Figure 5 Illustrating the core of a convex obstacle: the red points are corridor path terminals.

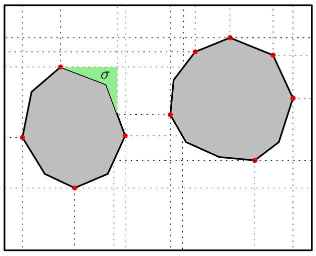

Figure 6 Illustrating the core-based cell decomposition $\mathcal{D}_{\mathcal{M}}$: the red vertices are core vertices and the green cell σ is a boundary cell.

consecutively by line segments to obtain another polygon, denoted by $core(P)$ and called the *core* of P (see Figure 5). Let $\mathcal{P}_{core}$ denote the complement of the union of all cores $core(P)$ for all $P \in \mathcal{B}$ and corridor paths in $\mathcal{P}$. Note that the number of vertices of $\mathcal{P}_{core}$ is $O(h)$ and $\mathcal{M} \subseteq \mathcal{P}_{core}$. For $s, t \in \mathcal{P}_{core}$, let $d_{core}(s,t)$ be the geodesic distance between s and t in $\mathcal{P}_{core}$.

The core structure leads to a more efficient way to find an L_1 shortest path between two points in $\mathcal{P}$. Chen and Wang [9] proved that an L_1 shortest path between $s, t \in \mathcal{M}$ in $\mathcal{P}_{core}$ can be locally modified to an L_1 shortest path in $\mathcal{P}$ without increasing its L_1 length.

▶ **Lemma 8** ([9]). *For any two points s and t in $\mathcal{M}$, $d(s,t) = d_{core}(s,t)$ holds.*

Hence, to compute $d(s,t)$ between two points s and t in $\mathcal{M}$, it is sufficient to consider only the cores and the corridor paths, that is, $\mathcal{P}_{core}$. We thus reduce the problem size from $O(n)$ to $O(h)$. Let $SPM_{core}(s)$ be a shortest path map for any source point $s \in \mathcal{M}$. Then, $SPM_{core}(s)$ has $O(h)$ complexity and can be computed in $O(h \log h)$ time [9].

We introduce a core-based cell decomposition $\mathcal{D}_{\mathcal{M}}$ of the ocean $\mathcal{M}$ (see Figure 6) in order to fully exploit the advantage of the core structure in designing algorithms computing the L_1 geodesic diameter and center. For any $P \in \mathcal{B}$, the vertices of $core(P)$ are called *core vertices*.

The construction of $\mathcal{D}_{\mathcal{M}}$ is analogous to that of $\mathcal{D}$ for $\mathcal{P}$. We first extend a horizontal line only from each *core vertex* until it hits $\partial\mathcal{M}$ to have a horizontal diagonal, and then extend a vertical line from each core vertex and each endpoint of the above horizontal diagonal. The resulting cell decomposition induced by the above diagonals is $\mathcal{D}_{\mathcal{M}}$. Hence, $\mathcal{D}_{\mathcal{M}}$ is constructed in $\mathcal{M}$ with respect to core vertices. Note that $\mathcal{D}_{\mathcal{M}}$ consists of $O(h^2)$ cells and can be built in $O(n \log n + h^2)$ time by a typical plane sweep algorithm. We call a cell σ of $\mathcal{D}_{\mathcal{M}}$ a *boundary cell* if $\partial\sigma \cap \partial\mathcal{M} \neq \emptyset$. For any boundary cell σ, the portion $\partial\sigma \cap \partial\mathcal{M}$ appears as a convex chain of $P \in \mathcal{B}$ by our construction of its core and $\mathcal{D}_{\mathcal{M}}$; since $\partial\sigma \cap \partial\mathcal{M}$ may contain multiple vertices of $\mathcal{M}$, the complexity of σ may not be constant. Any non-boundary cell of $\mathcal{D}_{\mathcal{M}}$ is a rectangle bounded by four diagonals. Each vertex of $\mathcal{D}_{\mathcal{M}}$ is either an endpoint of its diagonal or an intersection of two diagonals; thus, the number of vertices of $\mathcal{D}_{\mathcal{M}}$ is $O(h^2)$. Below we prove an analogue of Lemma 1 for the decomposition $\mathcal{D}_{\mathcal{M}}$ of $\mathcal{M}$. Let V_σ be the set of vertices of $\mathcal{D}_{\mathcal{M}}$ incident to σ. Note that $|V_\sigma| \leq 4$. We define the alignedness relation between two cells of $\mathcal{D}_{\mathcal{M}}$ analogously to that for $\mathcal{D}$. We then observe an analogy to Lemma 1.

▶ **Lemma 9.** *Let σ, σ' be any two cells of $\mathcal{D}_{\mathcal{M}}$. If they are aligned, then $d(s,t) = |\overline{st}|$ for any $s \in \sigma$ and $t \in \sigma'$; otherwise, there exists a shortest s-t path in $\mathcal{P}$ containing two vertices $v \in V_\sigma$ and $v' \in V_{\sigma'}$ with $d(s,t) = |\overline{sv}| + d(v,v') + |\overline{v't}|$.*

4 Improved Algorithms

In this section, we further explore the geometric structures and give more properties about our decomposition. These results, together with our results in Section 3, help us to give improved algorithms that compute the diameter and center, using a similar algorithmic framework as in Section 2.

4.1 The Cell-to-Cell Geodesic Distance Functions

Recall that our preliminary algorithms in Section 2 rely on the nice behavior of the cell-to-cell geodesic distance function: specifically, d restricted to $\sigma \times \sigma'$ for any two cells $\sigma, \sigma' \in \mathcal{D}$ is the lower envelope of $O(1)$ linear functions. We now have two different cell decompositions, $\mathcal{D}$ of $\mathcal{P}$ and $\mathcal{D}_{\mathcal{M}}$ of $\mathcal{M}$. Here, we observe analogues of Lemmas 1 and 9 for any two cells in $\mathcal{D} \cup \mathcal{D}_{\mathcal{M}}$, by extending the alignedness relation between cells in $\mathcal{D}$ and $\mathcal{D}_{\mathcal{M}}$, as follows.

Consider the geodesic distance function d restricted to $\sigma \times \sigma'$ for any two cells $\sigma, \sigma' \in \mathcal{D} \cup \mathcal{D}_{\mathcal{M}}$. We call a cell $\sigma \in \mathcal{D} \cup \mathcal{D}_{\mathcal{M}}$ *oceanic* if $\sigma \subset \mathcal{M}$, or *coastal*, otherwise. If both $\sigma, \sigma' \in \mathcal{D} \cup \mathcal{D}_{\mathcal{M}}$ are coastal, then $\sigma, \sigma' \in \mathcal{D}$ and the case is well understood as discussed in Section 2. Otherwise, there are two cases: the *ocean-to-ocean case* where both σ and σ' are oceanic, and the *coast-to-ocean case* where only one of them is oceanic.

For the ocean-to-ocean case, we extend the alignedness relation for all oceanic cells in $\mathcal{D} \cup \mathcal{D}_{\mathcal{M}}$. To this end, when both σ and σ' are in $\mathcal{D}$ or $\mathcal{D}_{\mathcal{M}}$, the alignedness has already been defined. For any two oceanic cells $\sigma \in \mathcal{D}$ and $\sigma' \in \mathcal{D}_{\mathcal{M}}$, we define their alignedness relation in the following way. If σ is contained in a cell $\sigma'' \in \mathcal{D}_{\mathcal{M}}$ that is aligned with σ', then we say that σ and σ' are *aligned*. However, σ may not be contained in a cell of $\mathcal{D}_{\mathcal{M}}$ because the endpoints of horizontal diagonals of $\mathcal{D}_{\mathcal{M}}$ that are on bay/canal gates are not vertices of $\mathcal{D}$ and those endpoints create vertical diagonals in $\mathcal{D}_{\mathcal{M}}$ that are not in $\mathcal{D}$. To resolve this issue, we augment $\mathcal{D}$ by adding the vertical diagonals of $\mathcal{D}_{\mathcal{M}}$ to $\mathcal{D}$. Specifically, for each vertical diagonal l of $\mathcal{D}_{\mathcal{M}}$, if no diagonal in $\mathcal{D}$ contains l, then we add l to $\mathcal{D}$ and extend l vertically until it hits the boundary of $\mathcal{P}$. In this way, we add $O(h)$ vertical diagonals to $\mathcal{D}$, and the size of $\mathcal{D}$ is still $O(n^2)$. Further, all results we obtained before are still applicable to the new $\mathcal{D}$. By abusing the notation, we still use $\mathcal{D}$ to denote the new version of $\mathcal{D}$. Now, for any two oceanic cells $\sigma \in \mathcal{D}$ and $\sigma' \in \mathcal{D}_{\mathcal{M}}$, there must be a unique cell $\sigma'' \in \mathcal{D}_{\mathcal{M}}$ that contains σ, and σ and σ' are defined to be *aligned* if and only if σ'' and σ' are aligned. Lemmas 1 and 9 are naturally extended as follows, along with this extended alignedness relation.

▶ **Lemma 10.** *Let $\sigma, \sigma' \in \mathcal{D} \cup \mathcal{D}_{\mathcal{M}}$ be two oceanic cells. For any $s \in \sigma$ and $t \in \sigma'$, it holds that $d(s,t) = |\overline{st}|$ if σ and σ' are aligned; otherwise, there exists a shortest s-t path that passes through a vertex $v \in V_\sigma$ and a vertex $v' \in V_{\sigma'}$.*

We then turn to the coast-to-ocean case. We now focus on a bay or canal A. Since A has gates, we need to somehow incorporate the influence of its gates into the decomposition $\mathcal{D}$. To this end, we add $O(1)$ additional diagonals into $\mathcal{D}_{\mathcal{M}}$ as follows: extend a horizontal line from each endpoint of each gate of A until it hits $\partial\mathcal{M}$, and then extend a vertical line from each endpoint of each gate of A and each endpoint of the horizontal diagonals that are added above. Let $\mathcal{D}^A_{\mathcal{M}}$ denote the resulting decomposition of $\mathcal{M}$. Note that there are some cells of $\mathcal{D}_{\mathcal{M}}$ each of which is partitioned into $O(1)$ cells of $\mathcal{D}^A_{\mathcal{M}}$ but the combinatorial complexity of $\mathcal{D}^A_{\mathcal{M}}$ is still $O(h^2)$. For any gate g of A, let $C_g \subset \mathcal{P}$ be the cross-shaped region of points in $\mathcal{P}$ that can be joined with a point on g by a vertical or horizontal line segment inside $\mathcal{P}$. Since the endpoints of g are also obstacle vertices, the boundary of C_g is formed by four diagonals of $\mathcal{D}$. Hence, any cell in $\mathcal{D}$ or $\mathcal{D}^A_{\mathcal{M}}$ is either completely contained in C_g or interior-disjoint from C_g. A cell of $\mathcal{D}$ or $\mathcal{D}^A_{\mathcal{M}}$ in the former case is said to be *g-aligned*.

In the following, we let $\sigma \in \mathcal{D}$ be any coastal cell that intersects A and $\sigma' \in \mathcal{D}^A_{\mathcal{M}}$ be any oceanic cell. Depending on whether σ and σ' are g-aligned for a gate g of A, there are three cases: (1) both cells are g-aligned; (2) σ' is not g-aligned; (3) σ' is g-aligned but σ is not. Lemma 11 handles the first case. Lemma 12 deals with a special case for the latter two cases. Lemma 13 is for the second case. Lemma 15 is for the third case and Lemma 14 is for proving Lemma 15. The proof of Lemma 16 summarizes the entire algorithm for all three cases.

▶ **Lemma 11.** *Suppose that σ and σ' are both g-aligned for a gate g of A. Then, for any $s \in \sigma$ and $t \in \sigma'$, we have $d(s,t) = |\overline{st}|$.*

Consider any path π in $\mathcal{P}$ from $s \in \sigma$ to $t \in \sigma'$, and assume π is directed from s to t. For a gate g of A, we call π *g-through* if g is the last gate of A crossed by π. The path π is a *shortest g-through path* if its L_1 length is the smallest among all g-through paths from s to t. Suppose π is a shortest path from s to t in $\mathcal{P}$. Since σ may intersect $\mathcal{M}$, if $s \in \sigma$ is not in A, then π may *avoid* A (i.e., π does not intersect A). If A is a bay, then either π avoids A or π is a shortest g-through path for the only gate g of A; otherwise (i.e., A is a canal), either π avoids A or π is a shortest g-through or g'-through path for the two gates g and g' of A. We have the following lemma, which is self-evident.

▶ **Lemma 12.** *Suppose that for any gate g of A, at least one of σ and σ' is not g-aligned. For any $s \in \sigma$ and $t \in \sigma'$, if there exists a shortest s-t path that avoids A, then a shortest s-t path passes through a vertex $v \in V_\sigma$ and another vertex $v' \in V_{\sigma'}$.*

We then focus on shortest g-through paths according to the g-alignedness of σ and σ'.

▶ **Lemma 13.** *Suppose σ' is not g-aligned for a gate g of A and there are no shortest s-t paths that avoid A. Then, for any $s \in \sigma$ and $t \in \sigma'$, there exists a shortest g-through s-t path containing a vertex $v \in V_\sigma$ and a vertex $v' \in V_{\sigma'}$.*

The remaining case is when $\sigma' \in \mathcal{D}^A_{\mathcal{M}}$ is g-aligned but $\sigma \in \mathcal{D}$ is not. Recall σ is coastal and intersects A, and σ' is oceanic (implying σ' does not intersect A).

▶ **Lemma 14.** *Let g be a gate of A, and suppose that σ is not g-aligned. Then, there exists a unique vertex $v_g \in V_\sigma \cap A$ such that for any $s \in \sigma$ and $x \in g$, the concatenation of segment $\overline{sv_g}$ and any L_1 shortest path from v_g to x inside $A \cup \sigma$ results in an L_1 shortest path from s to x in $A \cup \sigma$.*

From now on, let v_g be the vertex as described in Lemma 14 (v_g can be found efficiently, as shown in the proof of Lemma 16). Consider the union of the Euclidean shortest paths inside A from v_g to all points $x \in g$. Since A is a simple polygon, the union forms a funnel $F_g(v_g)$ with base g, plus the Euclidean shortest path from v_g to the apex of $F_g(v_g)$. Recall Fact 2 that any Euclidean shortest path inside a simple polygon is also an L_1 shortest path. Let $W_g(v_g)$ be the set of horizontally and vertically extreme points in each convex chain of $F_g(v_g)$, that is, $W_g(v_g)$ gathers the leftmost, rightmost, uppermost, and lowermost points in each chain of $F_g(v_g)$. Note that $|W_g(v_g)| \le 8$ and $W_g(v_g)$ includes the endpoints of g and the apex of $F_g(v_g)$. We then observe the following lemma.

▶ **Lemma 15.** *Suppose that σ' is g-aligned but σ is not. Then, for any $s \in \sigma$ and $t \in \sigma'$, there exists a shortest g-through s-t path that passes through v_g and some $w \in W_g(v_g)$. Moreover, the length of such a path is $|\overline{sv_g}| + d(v_g, w) + |\overline{wt}|$.*

Proof. Since A is a simple polygon, any Euclidean shortest path in A is also an L_1 shortest path by Fact 2. Thus, the L_1 length of a shortest path from v_g to any point x in the funnel $F_g(v_g)$ is equal to the L_1 length of the unique Euclidean shortest path in A, which is contained in $F_g(v_g)$.

By Lemma 14 and the assumption that σ' is g-aligned, among the paths from s to t that cross the gate g, there exists an L_1 shortest g-through s-t path π consisting of three portions: $\overline{sv_g}$, the unique Euclidean shortest path from v_g to a vertex u on a convex chain of $F_g(v_g)$, and $\overline{ut}$. Let $w \in W_g(v_g)$ be the last one among $W_g(v_g)$ that we encounter during the walk from s to t along π. Consider the segment $\overline{wt}$, which may cross $\partial F_g(v_g)$. If $\overline{wt} \cap \partial F_g(v_g) = \emptyset$, then we are done by replacing the subpath of π from u to t by $\overline{wt}$. Otherwise, $\overline{wt}$ crosses $\partial F_g(v_g)$ at two points $p, q \in \partial F_g(v_g)$. Since $W_g(v_g)$ includes all extreme points of each chain of $F_g(v_g)$, there is no $w' \in W_g(v_g)$ on the subchain of $F_g(v_g)$ between p and q. Hence, we can replace the subpath of π from w to t by a monotone path from w to t, which consists of $\overline{wp}$, the convex path from p to q along $\partial F_g(v_g)$, and $\overline{qt}$, and the L_1 length of the above monotone path is equal to $|\overline{wt}|$ by Fact 1. Consequently, the resulting path is also an L_1 shortest path with the desired property. ◀

For any cell $\sigma \in \mathcal{D} \cup \mathcal{D}_\mathcal{M}$, let n_σ be the size of σ. If σ is a boundary cell of $\mathcal{D}_\mathcal{M}$, then n_σ may not be bounded by a constant; otherwise, σ is a trapezoid or a triangle, and thus $n_\sigma \leq 4$. The geodesic distance function d defined on $\sigma \times \sigma'$ for any two cells $\sigma, \sigma' \in \mathcal{D} \cup \mathcal{D}_\mathcal{M}$ can be explicitly computed in $O(n_\sigma n_{\sigma'})$ time after some preprocessing, as shown in Lemma 16.

▶ **Lemma 16.** *Let σ be any cell of $\mathcal{D}$ or $\mathcal{D}_\mathcal{M}$. After $O(n)$-time preprocessing, the function d on $\sigma \times \sigma'$ for any cell $\sigma' \in \mathcal{D} \cup \mathcal{D}_\mathcal{M}$ can be explicitly computed in $O(n_\sigma n_{\sigma'})$ time, provided that $d(v, v')$ has been computed for any $v \in V_\sigma$ and any $v' \in V_{\sigma'}$. Moreover, d on $\sigma \times \sigma'$ is the lower envelope of $O(1)$ linear functions.*

Proof. If both σ and σ' are oceanic, then Lemma 10 implies that for any $(s, t) \in \sigma \times \sigma'$, $d(s, t) = |\overline{st}|$ if they are aligned, or $d(s, t) = \min_{v \in V_\sigma, v' \in V_{\sigma'}} d_{vv'}(s, t)$, where $d_{vv'}(s, t) = |\overline{sv}| + d(v, v') + |\overline{v't}|$. On the other hand, if σ and σ' are coastal, then both are cells of $\mathcal{D}$ and Lemma 1 implies the same conclusion. Since $|V_\sigma| \leq 4$ and $|V_{\sigma'}| \leq 4$ in either case, the geodesic distance d on $\sigma \times \sigma$ is the lower envelope of at most 16 linear functions. Hence, provided that the values of $d(v, v')$ for all pairs (v, v') are known, the envelope can be computed in time proportional to the complexity of the domain $\sigma \times \sigma'$, which is $O(n_\sigma n_{\sigma'})$.

From now on, suppose that σ is coastal and σ' is oceanic. Then, σ is a cell of $\mathcal{D}$ and intersects some bay or canal A. If σ' is also a cell of $\mathcal{D}$, then Lemma 1 implies the lemma, as discussed in Section 2; thus, we assume σ' is a cell of $\mathcal{D}_\mathcal{M}$.

As above, we add diagonals extended from each endpoint of each gate of A to obtain $\mathcal{D}^A_\mathcal{M}$, and specify all g-aligned cells for each gate g of A in $O(n)$ time. In the following, let σ' be an oceanic cell of $\mathcal{D}$ or of $\mathcal{D}^A_\mathcal{M}$. Note that a cell of $\mathcal{D}_\mathcal{M}$ can be partitioned into $O(1)$ cells of $\mathcal{D}^A_\mathcal{M}$. We have two cases whether A is a bay or a canal.

First, suppose that A is a bay; let g be the unique gate of A. In this case, any L_1 shortest path is g-through, provided that it intersects A, since g is unique. There are two subcases depending on whether σ is g-aligned or not.

- If σ is g-aligned, then by Lemmas 11, 12, and 13, we have $d(s, t) = |\overline{st}|$ if σ' is g-aligned, or $d(s, t) = \min_{v \in V_\sigma, v' \in V_{\sigma'}} d_{vv'}(s, t)$, otherwise, where $d_{vv'}(s, t) = |\overline{sv}| + d(v, v') + |\overline{v't}|$. Thus, the lemma follows by an identical argument as above.
- Suppose that σ is not g-aligned. Then, $\sigma \subset A$ since A has a unique gate g. In this case, we need to find the vertex $v_g \in V_\sigma$. For the purpose, we compute at most four Euclidean shortest path maps $SPM_A(v)$ inside A for all $v \in V_\sigma$ in $O(n)$ time [14]. By

Fact 2, $SPM_A(v)$ is also an L_1 shortest path map in A. We then specify the L_1 geodesic distance from v to all points on g, which results in a piecewise linear function f_v on g. For each $v \in V_\sigma$, we test whether it holds that $f_v(x) + |\overline{vv'}| \leq f_{v'}(x)$ for all $x \in g$ and all $v' \in V_\sigma$. By Lemma 14, there exists a vertex in V_σ for which the above test is passed, and such a vertex is v_g. Since each shortest path map $SPM_A(v)$ is of $O(n)$ complexity, all the above effort to find v_g is bounded by $O(n)$. Next, we compute the funnel $F_g(v_g)$ and the extreme vertices $W_g(v_g)$ as done above by exploring $SPM_A(v_g)$ in $O(n)$ time.

If σ' is not g-aligned, we apply Lemma 13 to obtain $d(s,t) = \min_{v \in V_\sigma, v' \in V_{\sigma'}} d_{vv'}(s,t)$. Thus, d is the lower envelope of at most 16 linear functions over $\sigma \times \sigma'$. Otherwise, if σ' is g-aligned, then we have $d(s,t) = \min_{w \in W_g(v_g)} d_{v_g w}(s,t)$ by Lemma 15. Since $|W_g(v_g)| \leq 8$, d is the lower envelope of a constant number of linear functions.

Thus, in any case, we conclude the bay case.

Now, suppose that A is a canal. Then, A has two gates g and g', and σ falls into one of the three case: (i) σ is both g-aligned and g'-aligned, (ii) σ is neither g-aligned nor g'-aligned, or (iii) σ is g- or g'-aligned but not both. As a preprocessing, if σ is not g-aligned, then we compute v_g, $F_g(v_g)$, and $W_g(v_g)$ as done in the bay case; analogously, if not g'-aligned, compute $v_{g'}$, $F_{g'}(v_{g'})$, and $W_{g'}(v_{g'})$. Note that any shortest path in $\mathcal{P}$ is either g-through or g'-through, provided that it intersects A. Thus, $d(s,t)$ chooses the minimum among a shortest g-through path, a shortest g'-through path, and a shortest path avoiding A if possible. We consider each of the three cases of σ.

1. Suppose that σ is both g-aligned and g'-aligned. In this case, if σ' is either g-aligned or g'-aligned, then we have $d(s,t) = |\overline{st}|$ by Lemma 11. Otherwise, if σ' is neither g-aligned nor g'-aligned, then we apply Lemmas 12 and 13 to have $d(s,t) = \min_{v \in V_\sigma, v' \in V_{\sigma'}} d_{vv'}(s,t)$. Hence, the lemma follows.
2. Suppose that σ is neither g-aligned nor g'-aligned. If σ' is both g-aligned and g'-aligned, then by Lemma 15 the length of a shortest g-through path is equal to $\min_{w \in W_g(v_g)} d_{v_g w}(s,t)$ while the length of a shortest g'-through path is equal to $\min_{w \in W_{g'}(v_{g'})} d_{v_{g'} w}(s,t)$. The geodesic distance $d(s,t)$ is the minimum of the above two quantities, and thus the lower envelope of $O(1)$ linear function on $\sigma \times \sigma'$.

   If σ' is g-aligned but not g'-aligned, then by Lemmas 13 and 15, we have
   $$d(s,t) = \min\{\min_{w \in W_g(v_g)} d_{v_g w}(s,t), \min_{v \in V_\sigma, v' \in V_{\sigma'}} d_{vv'}(s,t)\}.$$
   The case where σ' is g'-aligned but not g-aligned is analogous.

   If σ' is neither g-aligned nor g'-aligned, then $d(s,t) = \min_{v \in V_\sigma, v' \in V_{\sigma'}} d_{vv'}(s,t)$ by Lemma 13.
3. Suppose that σ is g'-aligned but not g-aligned. The other case where it is g-aligned but not g'-aligned can be handled symmetrically. If σ' is g'-aligned, then we have $d(s,t) = |\overline{st}|$ by Lemma 11. If σ' is neither g-aligned nor g'-aligned, then, by Lemmas 12 and 13, $d(s,t) = \min_{v \in V_\sigma, v' \in V_{\sigma'}} d_{vv'}(s,t)$.

   The remaining case is when σ' is g-aligned but not g'-aligned. In this case, the length of a shortest g-through path is equal to $\min_{w \in W_g(v_g)} d_{v_g w}(s,t)$ by Lemma 15 for gate g while the length of a shortest g'-through path is equal to $\min_{v \in V_\sigma, v' \in V_{\sigma'}} d_{vv'}(s,t)$ by Lemmas 12 and 13. Thus, the geodesic distance $d(s,t)$ is the smaller of the two quantities.

Consequently, we have verified every case of (σ, σ'). Finally, observe that it is sufficient to handle separately all the cells $\sigma' \in \mathcal{D}^A_\mathcal{M}$ whose union forms the original cell of $\mathcal{D}_\mathcal{M}$, since every cell of $\mathcal{D}_\mathcal{M}$ can be decomposed into $O(1)$ cells of $\mathcal{D}^A_\mathcal{M}$. ◀

4.2 Computing the Geodesic Diameter and Center

Lemma 7 assures that we can ignore coastal cells that are contained in the interior of a bay or canal, in order to find a farthest point from any $s \in \mathcal{P}$. This suggests a combined set $\mathcal{D}_f$ of cells from the two different decompositions $\mathcal{D}$ and $\mathcal{D}_\mathcal{M}$: Let $\mathcal{D}_f$ be the set of all cells σ such that either σ belongs to $\mathcal{D}_\mathcal{M}$ or $\sigma \in \mathcal{D}$ is a coastal cell with $\partial\sigma \cap \partial\mathcal{P} \neq \emptyset$. Note that $\mathcal{D}_f$ consists of $O(h^2)$ oceanic cells from $\mathcal{D}_\mathcal{M}$ and $O(n)$ coastal cells from $\mathcal{D}$. Since the boundary ∂A of any bay or canal A is covered by the cells of $\mathcal{D}_f$, Lemma 7 implies the following lemma.

▶ **Lemma 17.** *For any point $s \in \mathcal{P}$, $\max_{t\in\mathcal{P}} d(s,t) = \max_{\sigma'\in\mathcal{D}_f} \max_{t\in\sigma'} d(s,t)$.*

We apply the same approach as in Section 2 but we use $\mathcal{D}_f$ instead of $\mathcal{D}$.

To compute the diameter, we compute the (σ,σ')-constrained diameter for each pair of cells $\sigma,\sigma' \in \mathcal{D}_f$. Suppose we know the value of $d(v,v')$ for any $v \in V_\sigma$ and any $v \in V_{\sigma'}$ over all $\sigma,\sigma' \in \mathcal{D}_f$. Our algorithm handles each pair (σ,σ') of cells in $\mathcal{D}_f$ according to their types by applying Lemma 16. Lemma 18 computes $d(v,v')$ for all cell vertices v and v' of $\mathcal{D}_f$.

▶ **Lemma 18.** *In $O(n^2+h^4)$ time, one can compute the geodesic distances $d(v,v')$ between every $v \in V_\sigma$ and $v' \in V_{\sigma'}$ for all pairs of two cells $\sigma,\sigma' \in \mathcal{D}_f$.*

Our algorithms for computing the diameter and center are summarized in Theorem 19.

▶ **Theorem 19.** *The L_1 geodesic diameter and center of $\mathcal{P}$ can be computed in $O(n^2+h^4)$ and $O((n^4+n^2h^4)\alpha(n))$ time, respectively.*

Proof. We first discuss the diameter algorithm, whose correctness follows from Lemma 17.

After the execution of the procedure of Lemma 18 as a preprocessing, our algorithm considers three cases for two cells $\sigma,\sigma' \in \mathcal{D}_f$: (i) both are oceanic, (ii) both are coastal, or (iii) σ is coastal and σ' is oceanic. In either case, we apply Lemma 16.

For case (i), we have $O(h^2)$ oceanic cells and the total complexity is $\sum_{\sigma\in\mathcal{D}_\mathcal{M}} n_\sigma = O(n+h^2)$. Thus, the total time for case (i) is bounded by

$$\sum_{\sigma\in\mathcal{D}_\mathcal{M}} \sum_{\sigma'\in\mathcal{D}_\mathcal{M}} O(n_\sigma n_{\sigma'}) = \sum_{\sigma\in\mathcal{D}_\mathcal{M}} O(n_\sigma(n+h^2)) = O((n+h^2)^2) = O(n^2+h^4).$$

For case (ii), we have $O(n)$ coastal cells in $\mathcal{D}_\mathcal{M}$ and their total complexity is $O(n)$ since they are all trapezoidal. Thus, the total time for case (ii) is bounded by $O(n^2)$.

For case (iii), we fix a coastal cell $\sigma \in \mathcal{D}_f$ and iterates over all oceanic cells $\sigma' \in \mathcal{D}_\mathcal{M}$, after an $O(n)$-time preprocessing, as done in the proof of Lemma 16. For each σ, we take $O(n+h^2)$ time since $\sum_{\sigma\in\mathcal{D}_\mathcal{M}} n_\sigma = O(n+h^2)$. Thus, the total time for case (iii) is bounded by $O(n^2+nh^2) = O((n+h^2)^2) = O(n^2+h^4)$.

Next, we discuss our algorithm for computing a geodesic center of $\mathcal{P}$. We consider $O(n^2)$ cells $\sigma \in \mathcal{D}$ and compute all the σ-constrained centers. As a preprocessing, we spend $O(n^4)$ time to compute the geodesic distances $d(v,v')$ for all pairs of vertices of $\mathcal{D}$ by Lemma 3. Fix a cell $\sigma \in \mathcal{D}$. For all $\sigma' \in \mathcal{D}_f$, we compute the geodesic distance function d restricted to $\sigma \times \sigma'$ by applying Lemma 16. As in Section 2, compute the graph of $R_{\sigma'}(q) = \max_{p\in\sigma'} d(p,q)$ by projecting the graph of d over $\sigma \times \sigma'$, and take the upper envelope of the graphs of $R_{\sigma'}$ for all $\sigma' \in \mathcal{D}_f$. By Lemma 16, we have an analogue of Lemma 5 and thus a σ-constrained center can be computed in $O(m^2\alpha(m))$ time, where m denotes the total complexity of all $R_{\sigma'}$. Lemma 16 implies that $m = O(n+h^2)$.

For the time complexity, note that $\sum_{\sigma\in\mathcal{D}_\mathcal{M}} n_\sigma = O(n+h^2)$ and $\sum_{\sigma\in\mathcal{D}_f\setminus\mathcal{D}_\mathcal{M}} n_\sigma = O(n)$. Since each cell in $\mathcal{D}$ is either a triangle or a trapezoid, its complexity is $O(1)$. Thus, for each

$\sigma \in \mathcal{D}$, by Lemma 16, computing a σ-constrained center takes $O((n+h^2)^2\alpha(n))$ time, after an $O(n^4)$-time preprocessing (Lemma 3). Iterating over all $\sigma \in \mathcal{D}$ takes $O(n^2(n+h^2)^2\alpha(n)) = O((n^4+n^2h^4)\alpha(n))$ time. ◀

Acknowledgements. Work by S. W. Bae was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2013R1A1A1A05006927), and by the Ministry of Education (2015R1D1A1A01057220). M. Korman was supported in part by the ELC project (MEXT KAKENHI No. 24106008). J. Mitchell acknowledges support from the US-Israel Binational Science Foundation (grant 2010074) and the National Science Foundation (CCF-1018388, CCF-1526406). Y. Okamoto is partially supported by Grant-in-Aid for Scientific Research (KAKENHI) 24106005, 24700008, 24220003, 15K00009. V. Polishchuk is supported in part by Grant 2014-03476 from the Sweden's innovation agency VINNOVA. H. Wang was supported in part by the National Science Foundation (CCF-1317143).

References

1 H.-K. Ahn, L. Barba, P. Bose, J.-L. De Carufel, M. Korman, and E. Oh. A linear-time algorithm for the geodesic center of a simple polygon. In *Proc. of the 31st Symposium on Computational Geometry (SoCG)*, pages 209–223, 2015.

2 T. Asano and G. Toussaint. Computing the geodesic center of a simple polygon. Technical Report SOCS-85.32, McGill University, Montreal, Canada, 1985.

3 S.W. Bae, M. Korman, J.S.B. Mitchell, Y. Okamoto, V. Polishchuk, and H. Wang. Computing the L_1 geodesic diameter and center of a polygonal domain. arXiv:1512.07160, 2015.

4 S.W. Bae, M. Korman, and Y. Okamoto. The geodesic diameter of polygonal domains. *Discrete and Computational Geometry*, 50:306–329, 2013.

5 S.W. Bae, M. Korman, and Y. Okamoto. Computing the geodesic centers of a polygonal domain. In *Proc. of the 26th Canadian Conference on Computational Geometry*, 2014.

6 S.W. Bae, M. Korman, Y. Okamoto, and H. Wang. Computing the L_1 geodesic diameter and center of a simple polygon in linear time. *Computational Geometry: Theory and Applications*, 48:495–505, 2015.

7 R. Bar-Yehuda and B. Chazelle. Triangulating disjoint Jordan chains. *International Journal of Computational Geometry and Applications*, 4(4):475–481, 1994.

8 B. Chazelle. A theorem on polygon cutting with applications. In *Proc. of the 23rd Annual Symposium on Foundations of Computer Science*, pages 339–349, 1982.

9 D.Z. Chen and H. Wang. A nearly optimal algorithm for finding L_1 shortest paths among polygonal obstacles in the plane. In *Proc. of the 19th European Symposium on Algorithms*, pages 481–492, 2011.

10 D.Z. Chen and H. Wang. Computing the visibility polygon of an island in a polygonal domain. In *Proc. of the 39th International Colloquium on Automata, Languages and Programming*, pages 218–229, 2012. Journal version published online in *Algorithmica*, 2015.

11 D.Z. Chen and H. Wang. L_1 shortest path queries among polygonal obstacles in the plane. In *Proc. of the 30th Symposium on Theoretical Aspects of Computer Science*, pages 293–304, 2013.

12 D.Z. Chen and H. Wang. Visibility and ray shooting queries in polygonal domains. *Computational Geometry: Theory and Applications*, 48:31–41, 2015.

13 H. Edelsbrunner, L.J. Guibas, and M. Sharir. The upper envelope of piecewise linear functions: Algorithms and applications. *Discrete and Computational Geometry*, 4:311–336, 1989.

14 L.J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R.E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2(1-4):209–233, 1987.

15 J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Computational Geometry: Theory and Applications*, 4(2):63–97, 1994.

16 J. Hershberger and S. Suri. Matrix searching with the shortest-path metric. *SIAM Journal on Computing*, 26(6):1612–1634, 1997.

17 R. Inkulu and S. Kapoor. Planar rectilinear shortest path computation using corridors. *Computational Geometry: Theory and Applications*, 42(9):873–884, 2009.

18 S. Kapoor, S.N. Maheshwari, and J.S.B. Mitchell. An efficient algorithm for Euclidean shortest paths among polygonal obstacles in the plane. *Discrete and Computational Geometry*, 18(4):377–383, 1997.

19 J.S.B. Mitchell. An optimal algorithm for shortest rectilinear paths among obstacles. In *the 1st Canadian Conference on Computational Geometry*, 1989.

20 J.S.B. Mitchell. L_1 shortest paths among polygonal obstacles in the plane. *Algorithmica*, 8(1):55–88, 1992.

21 R. Pollack, M. Sharir, and G. Rote. Computing the geodesic center of a simple polygon. *Discrete and Computational Geometry*, 4(1):611–626, 1989.

22 S. Schuierer. Computing the L_1-diameter and center of a simple rectilinear polygon. In *Proc. of the International Conference on Computing and Information*, pages 214–229, 1994.

23 S. Suri. Computing geodesic furthest neighbors in simple polygons. *Journal of Computer and System Sciences*, 39:220–235, 1989.