Min-Link Shortest Path Maps and Fréchet Distance[†]

Atlas F. Cook IV[‡] Carola Wenk[‡]

Abstract

We present shortest path maps that support min-link queries from any source point on a *line segment* to destination points in either a simple polygon or a polygonal domain (a polygon with holes). These structures are more versatile than the traditional shortest path map from a fixed source *point*, and we use them to compute the min-link Fréchet distance between two polygonal curves. In addition to developing exact algorithms for min-link shortest path maps, the Fréchet distance, and the Hausdorff distance, we also present approximation algorithms that are accurate to within 1, 2, or 3 links of the optimal solution.

The min-link Fréchet distance between polygonal curves inside a simple polygon interprets the boundary of the simple polygon as an obstacle. We prove that a free space cell (i.e., the parameter space representing all min-link distances between a pair of line segments) has the property that all min-link distances less than or equal to some threshold value ε define a *free space* that need not be convex but is always x-monotone, y-monotone, and connected. We use this property plus a novel *additive $\mathcal{B}$ approximation* for the Fréchet distance that runs in *subquadratic* $O(\frac{kN}{\mathcal{B}} + \frac{N^2}{\mathcal{B}^2})$ time to compute the *exact* Fréchet distance in $O(kN + N^2)$ time. Here, N is the complexity of the polygonal curves, and k is the complexity of the simple polygon. Note that the exact runtime is asymptotically competitive with the $O(N^2 \log N)$ runtime of the traditional (no obstacles) Fréchet distance [4]. For a polygonal domain, the monotonicity and connectedness properties no longer hold, so we use our shortest path map structure to represent a free space cell.

1. Introduction

The comparison of geometric shapes is essential in various applications including computer vision, computer aided design, robotics, medical imaging, and drug design. The Fréchet distance is a similarity metric for continuous shapes that is defined using reparametrizations of the shapes. It is generally a more appropriate distance measure for continuous shapes than the often used Hausdorff distance.

A *min-link* path $\pi(s,t)$ is a polygonal path between two points $s,t \in \mathbb{R}^l$ that avoids a set of obstacles and has the fewest possible links. Min-link paths are fundamentally different from traditional shortest paths that measure length via an L_p metric, and min-link paths have a wealth of possible applications including robotic motion, wireless communications, geographic information systems, VLSI, computer vision, solid modeling, image processing, and even water pipe placement. These applications benefit from min-link paths because turns are costly while straight line movements are inexpensive. See [14] for an excellent survey of min-link paths.

Although similarity metrics have traditionally been applied in obstacle-free environments, recent works [8, 11, 15] have calculated similarity metrics using Euclidean geodesics. By contrast, this work focuses on *min-link* similarity metrics. The motivation for these studies is that Euclidean geodesics and min-link paths are natural distance measures for complex environments such as simple polygons, polygonal domains (polygons with holes), and surfaces such as terrains.

Min-link paths from a fixed-source in a *simple polygon* of $O(k)$ complexity are studied by Suri [18] who extends the Euclidean shortest paths approach of Guibas et al. [13] to construct a *window partition* in linear time. This window partition is essentially a shortest path map because it divides the simple polygon into regions of equal link distance from a fixed-source, where the fixed-source is either a point or a line segment. Note that queries from a fixed-source *line segment* $\overline{ab}$ to a query point t always return $\min_{s\in\overline{ab}} d(s,t)$ (i.e., the shortest distance from the query point to any point on $\overline{ab}$). Queries from a particular point $s \in \overline{ab}$ are not

[†]This work has been supported by the National Science Foundation grant NSF CAREER CCF-0643597.

[‡]Department of Computer Science, University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249-0667. {acook,carola}@cs.utsa.edu

supported. By contrast, the work of Arkin, Mitchell, and Suri [6] supports $O(\log k)$ time queries between any two points in a simple polygon after building $O(k^2)$ shortest path maps with $\Theta(k^3)$ time and space preprocessing. These bounds have been matched in [12] by a planesweep approach.

Min-link paths in a *polygonal domain* (a polygon with holes) of $O(k)$ complexity have been studied by Mitchell, Rote, and Woeginger [17]. Their shortest path map structure supports fixed-source min-link queries in a polygonal domain in $O(\log k)$ time after $\Theta(k^4)$ time and space preprocessing. A related *Euclidean* result by Chiang and Mitchell [10] constructs a Euclidean two-point shortest path map from any source point in the plane to any target point in the plane in $O(k^{11})$ time and space.

Min-link problems are usually more difficult to solve than equivalent Euclidean shortest path problems because optimal paths that are unique under the Euclidean metric need not be unique under the min-link distance. Another difficulty is that Euclidean shortest paths only turn at obstacle vertices while min-link paths can turn anywhere [6, 14].

While previous work has focused on *Euclidean* problems such as shortest path queries [10, 13] and the geodesic Fréchet distance [8, 11, 15], we present the first algorithms for several *min-link* problems in a simple polygon or polygonal domain. Min-link shortest path maps are developed that support queries from any specified point on a source line, and these structures are applied to solve the min-link Fréchet distance. We also show how to compute the min-link Hausdorff distance for sets of points or sets of line segments. In addition to these exact algorithms, additive approximation algorithms that are accurate to within 1, 2, or 3 links of the optimal solution are presented for most of these problems. One of these additive approximations is the first *subquadratic* approximation for the Fréchet distance. It works by bundling free space cells together to obtain an *additive $\mathcal{B}$ approximation* in $O(\frac{kN}{\mathcal{B}} + \frac{N^2}{\mathcal{B}^2})$ time. Furthermore, this approximation can be applied to solve the *exact* min-link Fréchet distance in a simple polygon in only $O(kN + N^2)$ time. A few *Euclidean* problems are also considered such as the geodesic Hausdorff distance of point sets and line segments in a polygonal domain.

2. PRELIMINARIES

Hausdorff distance is a similarity metric commonly used to compare sets of points or sets of higher dimensional objects such as line segments or triangles. The *directed* Hausdorff distance is defined as $\tilde{\delta}_H(A, B) = \sup_{a \in A} \inf_{b \in B} d'(a, b)$, where A and B are compact sets and d' is a distance metric for points, usually the L_2 distance, and in our setting the min-link distance (see [4, 5]). The (*undirected*) geodesic Hausdorff distance is the larger of the two directed distances: $\delta_H(A, B) = \max(\tilde{\delta}_H(A, B),\ \tilde{\delta}_H(B, A))$.

The *Fréchet distance* for two curves $A, B : [0, 1] \to \mathbb{R}^l$ is defined as

$$\delta_F(A, B) = \inf_{f,g:[0,1]\to[0,1]} \sup_{t\in[0,1]} d'(\ A(f(t)), B(g(t))\ )$$

where f and g range over continuous non-decreasing reparametrizations. For a given $\varepsilon > 0$ the *free space* is defined as $FS_\varepsilon(A, B) = \{(s, t) \mid d'(A(s), B(t)) \leq \varepsilon\} \subseteq [0, 1]^2$. A free space cell $C \subseteq [0, 1]^2$ is the parameter space defined by two line segments $\overline{ab} \in A$ and $\overline{cd} \in B$, and the free space inside the cell is $FS_\varepsilon(\overline{ab}, \overline{cd}) = FS_\varepsilon(A, B) \cap C$.

The *decision problem* to check whether the Fréchet distance of two polygonal curves is at most a given parameter $\varepsilon > 0$ is solved by Alt and Godau [4] using a *free space diagram* which consists of all free space cells for all pairs of line segments of A and B. Their dynamic programming algorithm checks for the existence of a monotone path[1] in the free space from $(0, 0)$ to $(1, 1)$ by propagating reachability information cell by cell through the free space. *Reachable space* is the set of free space points that are reachable by a monotone path through the free space that originates at $(0, 0)$.

Let k be the complexity of a set of obstacles, and let N be the maximum of the complexities of two polygonal curves A and B. A *min-link* path $\pi(s, t)$ is a polygonal path between two points $s, t \in \mathbb{R}^l$ that avoids a set of obstacles and has the fewest possible links. Let $d(s, t)$ be the number of links on $\pi(s, t)$.

[1] A *monotone* path in the free space corresponds to a non-decreasing reparametrization of the polygonal curves A and B.

Let $\pi_E(s,t)$ and $d_E(s,t)$, respectively, represent the *Euclidean* shortest path and shortest path distance between s and t. Denote by $\alpha(N)$ the extremely slowly growing inverse Ackermann function. A function is called *bitonic* when it has at most one change in monotonicity, and a function is called "↓↑-bitonic" when it decreases monotonically then increases monotonically. The below observation shows that a set of points in a simple polygon P can be connected by a polygonal path by basically traversing the boundary of P after point location has been used to associate each point with one directly visible vertex of P.

Observation 1. *Any set $\mathcal{S}$ of $O(N)$ points in a simple polygon can be connected by a polygonal path with $O(k+N)$ links that stays inside the simple polygon. This path can be computed in $O(k+N\log k)$ time and $O(k+N)$ space, and there are instances in which $\Omega(k+N)$ links are necessary.*

Proof. The optimal asymptotic number of links for any polygonal path connecting $O(N)$ arbitrary points in a simple polygon is $\Omega(k+N)$ because (1) a path connecting just two points in a simple polygon can have length $O(k)$ and (2) a path connecting $O(N)$ points must in general have $O(N)$ links. Such a path can be constructed as follows. Triangulate the simple polygon in $O(k)$ time such that every vertex in the triangulation is a vertex of the simple polygon [9]. For each point $p \in \mathcal{S}$, use point location to locate the triangle containing p and associate p with one of the three vertices of its triangle. This takes $O(N\log k)$ total time. To construct a polygonal path of length $O(k+N)$ that visits each $p \in \mathcal{S}$, visit the vertices of the simple polygon in clockwise order. At each vertex v, connect v to the previous vertex on the simple polygon and connect v to each $p \in \mathcal{S}$ that is associated with v by adding the links $\overline{vp}$ and $\overline{pv}$ to the path. □

3. ALGORITHMS

3.1. **Shortest Path Maps.** A traditional shortest path map is a partition of the plane into a set of faces such that all points in a given face have the same combinatorial shortest path to a fixed source [16]. This means that all points in a given face have the same shortest path except possibly for the start and end points of the path. When the fixed source in a traditional shortest path map is a line segment $\overline{ab}$, the distance to a query point t is returned as $\min_{s\in\overline{ab}} d(s,t)$. By contrast, we develop min-link shortest path maps from a line segment $\overline{ab}$ that support queries from any desired source point $s \in \overline{ab}$ to a destination point in logarithmic query time. Approximation algorithms accurate to within 1, 2, and 3 links are also given. Note that these approaches can easily be adapted to handle a source that is an infinite *line*.

We use the following terminology of Arkin, Mitchell, and Suri [6]: the combinatorial type of a *shortest path map* is a listing of the combinatorial types of its edges. The combinatorial type of a shortest path map *edge* $\mathcal{E}$ is a vertex-edge pair (v,e) such that $\mathcal{E}$ has one endpoint at an obstacle vertex v and has its other endpoint on an obstacle edge e. The idea is that as the source point s varies along $\overline{ab}$, the position of $\mathcal{E}$'s endpoint on e is parametrized homographically by $g(s) = \frac{A+Bs}{C+Ds}$ (see Figure 3.1a). Furthermore, for any min-link path $\pi(s,t)$ there is a *representative* min-link path $\tilde{\pi}(s,t)$ such that all links except the final link of $\tilde{\pi}(s,t)$ overlap a shortest path map edge and touch a vertex (see Figure 3.1b).

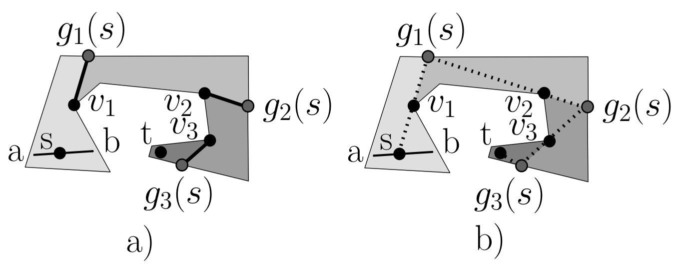

FIGURE 3.1. a) Every shortest path map edge can be described as a vertex-edge pair. b) All links except the final link of $\tilde{\pi}(s,t)$ overlap a shortest path map edge.

The approach of [6] is to answer min-link queries in a simple polygon P by partitioning the boundary of P into $O(k^2)$ *atomic segments*. An atomic segment $\overline{\alpha\beta}$ is a line segment such that the shortest path map for every source point $s \in \overline{\alpha\beta}$ has the same combinatorial type. By computing a representative shortest path map for each atomic segment on the boundary of P (and using a back-projection technique), optimal time min-link distance and path queries can be supported from any $s \in \mathbb{R}^2$ to any $t \in \mathbb{R}^2$ after $\Theta(k^3)$ preprocessing [12].

The atomic segments of [6] are computed by intersecting the $O(k)$ shortest path maps defined by each obstacle vertex $w \in P$ with the boundary of P. The result is $O(k^2)$ edges that partition the boundary of P into $O(k^2)$ atomic segments, where each atomic segment endpoint s' is the intersection of some edge of SPM(w) with the boundary of P.

The idea behind this construction is that the homography describing an edge remains valid until that homography is interrupted – either when $g(s)$ reaches an endpoint of e or when line of sight from v to $g(s)$ is blocked by the endpoint of an edge e'. In both cases, for a shortest path map edge $\mathcal{E}$ to change combinatorially at $s' \in \overline{ab}$, $\overline{vg(s')}$ must touch some vertex w. When this occurs, the distance from w to $\overline{ab}$ changes at s', so an edge $\mathcal{E}$ of SPM(w) must intersect $\overline{ab}$ at s'. Hence, $\mathcal{E}$ can only change combinatorially at the endpoint of an atomic segment.

Theorem 1. *A min-link shortest path map SPM($\overline{ab}$,$\mathbb{R}^2$) in a* simple polygon P *with* $O(k)$ *complexity can be constructed in* $O(k^2)$ *time and space such that the min-link distance* $d(s,t)$ *for any* $s \in \overline{ab}$ *and* $t \in \mathbb{R}^2$ *can be returned in* $O(\log k)$ *time, and* $\pi(s,t)$ *can be returned in* $O(\log k + K)$ *time, where* K *is the complexity of* $\pi(s,t)$. *In addition, an* approximate *SPM($\overline{ab}$,$\mathbb{R}^2$) that provides the same query times and is accurate to within* one *link of the exact distance can be built in* $O(k)$ *time and space.*

Proof. We partition $\overline{ab}$ into $O(k)$ atomic segments $\overline{\alpha_1\beta_1}, ..., \overline{\alpha_{O(k)}\beta_{O(k)}}$ by computing $\overline{ab} \cap$ SPM(w) for each obstacle vertex $w \in P$. Each of these intersections produces at most three atomic segments on $\overline{ab}$ (see [6]).

A query $d(s,t)$ for SPM($\overline{ab}$,$\mathbb{R}^2$) consists of a one dimensional coordinate for s and a two dimensional coordinate for $t \in P$. To support point location in this three dimensional space, we construct a modified SPM(α_i) for each atomic segment endpoint $\alpha_1, ..., \alpha_{O(k)}$. By parametrizing the edges of each SPM(α_i) based on the position of $s \in \overline{\alpha_i\beta_i}$, all min-link queries for any fixed $s \in \overline{\alpha_i\beta_i}$ can be handled by SPM(α_i). To speed up the query process, a triangulation is computed for SPM(α_i) such that at most two parametrized edges need to be evaluated at query time.

By section 3.1, SPM(α_i) contains $O(k)$ parametrized edges that are represented as vertex-edge pairs (v,e). By [6], over *all* $s \in \overline{\alpha_i\beta_i}$ there are at most two parametrized shortest path map edges (v_1,e), $(v_2,e) \in$ SPM(α_i) that can intersect the interior of a fixed boundary edge $e \in P$. As s varies from α_i to β_i, $(v_1,e) \cap e$ is a subsegment $\overline{\sigma\varsigma} \in e$ such that the three points v_1, σ, and ς define a triangle Δ_1. Similarly, the parametrization for (v_2,e) defines a second triangle Δ_2. By triangulating $\Delta_1 \cup \Delta_2$, we obtain a constant number of triangles that can be associated with (v_1,e),(v_2,e). No other parametrized shortest path map edges can intersect these triangles for any $s \in \overline{\alpha_i\beta_i}$ because shortest path map edges never cross each other in a simple polygon. After processing all shortest path map edges in this fashion, any regions in P not yet covered by a triangle are independent of the position of $s \in \overline{\alpha_i\beta_i}$ and can be triangulated in standard fashion.

A query $d(s,t)$ proceeds as follows. s lies in an atomic segment $\overline{\alpha_i\beta_i}$ that can be identified by binary search. Point location in SPM(α_i) identifies the triangle containing t in $O(\log k)$ time, and the at most two parametrized edges associated with this triangle can be evaluated in constant time to determine $d(s,t)$. $\pi(s,t)$ is calculated by following a chain of predecessors.

Approximate min-link distance queries from $s \in \overline{ab}$ to $t \in \mathbb{R}^2$ can be answered after computing a *single* shortest path map from the fixed-source $\overline{ab}$ in $O(k)$ time and space (see [18]). Although this shortest path map does not have the ability to return $d(s,t)$, it does have the ability to return the shortest distance from a query point t to its nearest point $s' \in \overline{ab}$. By the definition of min-link paths, all min-link distances from t to $\overline{ab}$ differ by at most one, so $d(s,s')+d(s',t)$ equals either $d(s,t)$ or $d(s,t)+1$. The approximate path is $\pi(s,s') \circ \pi(s',t)$, where $\circ$ denotes concatenation of polygonal paths. □

Although a min-link shortest path map SPM($\mathcal{S}$) from a fixed source $\mathcal{S}$ in a *polygonal domain* has $\Theta(k^4)$ complexity [17], we show next that SPM($\mathcal{S}$)$\cap\overline{cd}$ has only quadratic complexity and can be built in near-quadratic time. This property will be useful for computing shortest path maps and applying them to the Fréchet distance.

Lemma 1. *A min-link shortest path map SPM($\mathcal{S}$) from a point or line segment source $\mathcal{S}$ in a* polygonal domain *intersects a line segment $\overline{cd}$ in at most $O(k^2)$ intervals. These intervals can be constructed in $O(k^2\alpha(k)\log^2 k)$ time and $O(k^2)$ space.*

Proof. Suri and O'Rourke [19] have proven that all points in a polygonal domain with line of sight to a source line segment can be represented by the union of $O(k^2)$ triangles. Mitchell, Rote, and Woeginger [17] extend this idea to represent all points with min-link distance j to $\mathcal{S}$ by the union of $O(k^2)$ triangles.

To compute the intersection of SPM($\mathcal{S}$) with $\overline{cd}$, first compute $j = d(s, c)$ for any $s \in \mathcal{S}$ in $O(k^2\alpha(k)\log^2 k)$ time and $O(k^2)$ space using the algorithm of [17]. By the definition of min-link paths, distances from $\mathcal{S}$ to $\overline{cd}$ equal either i, $i+1$, or $i+2$, where $i = \min_{s\in\mathcal{S},\ t\in\overline{cd}} d(s, t)$. Hence, we are guaranteed that only paths of length $j-2$, $j-1$, j, $j+1$, or $j+2$ can intersect $\overline{cd}$. $O(k^2)$ triangles are sufficient to represent the union of these points, and these triangles can be constructed in $O(k^2\alpha(k)\log^2 k)$ time and $O(k^2)$ space using the algorithm of [17]. □

Both Euclidean shortest path and min-link problems in a polygonal domain of k complexity are often quite costly to solve. For example, Chiang and Mitchell [10] construct a *Euclidean* two-point shortest path map from any source point in the plane to any target point in the plane in $O(k^{11})$ time and space, and Mitchell, Rote, and Woeginger [17] construct a *min-link* shortest path map from a fixed source in $\Theta(k^4)$ time and space. Theorem 2 presents the first min-link shortest path map in a polygonal domain that supports queries from a continuous set of source points.

Theorem 2. *A min-link shortest path map SPM($\overline{ab}$,$\overline{cd}$) in a polygonal domain can be constructed in $O(k^7)$ expected time and space such that the min-link distance $d(s, t)$ for any $s \in \overline{ab}$ and $t \in \overline{cd}$ can be returned in $O(\log k)$ time, and $\pi(s, t)$ can be returned in $O(\log k + K)$ time, where K is the complexity of $\pi(s, t)$.* Approximate *queries for SPM($\overline{ab}$,$\mathbb{R}^2$) with $s \in \overline{ab}$ and $t \in \mathbb{R}^2$ are supported to within* one *link of optimal after $O(k^4)$ time and space preprocessing and to within* two *links of optimal after $O(k^{\frac{7}{3}}\log^{3.11} k)$ time and $O(k)$ space preprocessing.*

Proof. A set of atomic segments on $\overline{ab}$ can be defined (as in the simple polygon case) by computing a shortest path map from each obstacle vertex and intersecting the edges in these structures with $\overline{ab}$ [6]. Although each of the shortest path map structures for the $O(k)$ obstacle vertices has $\Theta(k^4)$ complexity [17, 19], Lemma 1 ensures that there are only $O(k\cdot k^2)$ atomic segments $\overline{\alpha_1\beta_1}, ..., \overline{\alpha_R\beta_R} \in \overline{ab}$ and that for each atomic segment $\overline{\alpha_i\beta_i} \in \overline{ab}$, SPM($\alpha_i$) $\cap$ $\overline{cd}$ can be computed in $O(k^2\alpha(k)\log^2 k)$ time and $O(k^2)$ space.

We construct SPM($\overline{ab}$,$\overline{cd}$) as a partition of $\overline{ab} \times \overline{cd}$. For each atomic segment $\overline{\alpha_i\beta_i}$, a shortest path map edge $\mathcal{E} \in$ SPM(α_i) is described by a vertex-edge pair (v, e) and can be homographically parametrized such that $\mathcal{E}\cap\overline{cd}$ defines a constant complexity algebraic curve in $\overline{\alpha_i\beta_i}\times\overline{cd}$ as s varies from α_i to β_i. Constructing such a curve for each choice of v and e yields $O(k^2)$ curves whose arrangement can be constructed in $O(k^4)$ expected time and space. Hence, the partition of $\overline{\alpha_i\beta_i} \times \overline{cd}$ into faces that encode combinatorial shortest paths has $O(k^4)$ complexity. Since there are $O(k^3)$ atomic segments, SPM($\overline{ab}$,$\overline{cd}$) has $O(k^3\cdot k^4)$ complexity. A randomized incremental algorithm (see [2]) can build both the arrangements and a point location structure in $O(k^7)$ expected time.

Approximate min-link queries from $s \in \overline{ab}$ to $t \in \mathbb{R}^2$ that are accurate to within *one* link of the exact distance are available by computing a single shortest path map from the fixed-source $\overline{ab}$ in $O(k^4)$ time and space and using the shortest distance $d(s', t)$ from a query point t to its nearest point $s' \in \overline{ab}$ [17]. Queries accurate to within *two* links can be found after $O(k^{\frac{7}{3}}\log^{3.11} k)$ time and $O(k)$ space preprocessing by allowing $d(s', t)$ to be computed approximately to within one link of optimal [17]. □

3.2. **Hausdorff Distance for Point Sets.** The *Hausdorff distance* is a similarity metric used to compare two compact sets (see section 2).

Theorem 3. *The* min-link *Hausdorff distance between point sets can be computed in $O(kN + N^2)$ time and $O(k + N)$ space in a simple polygon and in $O(Nk^{\frac{7}{3}} \log^{3.11} k + N^2 k)$ time and $O(k + N)$ space in a polygonal domain.*[2]

Proof. For min-link queries from a fixed source in a simple polygon, Suri [18] has shown how to construct a shortest path map in $O(k)$ time and space. Although it is straightforward to achieve $O(kN + N^2 \log k)$ time for the Hausdorff distance, we can shave off the $O(\log k)$ factor as follows. Preprocess the $O(N)$ points in B into a polygonal path with length $O(k + N)$. By Observation 1, this polygonal path can be constructed *once* in $O(k + N \log k)$ time and $O(k + N)$ space. Since any line segment intersects at most three faces of Suri's shortest path map (see [6]), the distance $\min_{b \in B} d(a, b)$ can be returned in $O(k + N)$ time by constructing a shortest path map SPM(a) in $O(k)$ time and walking through at most three faces of SPM(a) for each line segment of the polygonal path in $O(k + N)$ time. Hence, the Hausdorff distance can be computed in $O(N \cdot (k + N))$ time.

For min-link distances in a polygonal domain, $O(N)$ shortest path map structures due to Mitchell, Rote, and Woeginger [17] can be built in $O(Nk^{\frac{7}{3}} \log^{3.11} k)$ time such that each of the $O(N^2)$ distance queries can be answered in $O(k)$ time. The space bounds follow for all three algorithms by storing only the points in $A \cup B$ and a single shortest path map at a time. □

3.3. **Hausdorff Distance for Line Segment Sets.** The standard approach [3] to compute the Hausdorff distance is to use Voronoi diagrams. For domains where no efficient Voronoi diagrams are available, the Hausdorff distance between line segment sets A and B can be computed using lower envelopes. Define $f_{\overline{ab}} : [a, b] \to \mathbb{R}$ with $f_{\overline{ab}}(s) = \min_{t \in B} d(s, t)$. $f_{\overline{ab}}$ represents the nearest neighbor for every $s \in \overline{ab}$ to any point in B and equals the lower envelope of $O(N)$ simpler functions $f_{\overline{ab},\, \overline{cd}} : [a, b] \to \mathbb{R}$ such that $f_{\overline{ab},\, \overline{cd}}(s) = \min_{t \in \overline{cd}} d(s, t)$ (note that there is one such function for each choice of $\overline{cd} \in B$). The Hausdorff distance can be calculated by computing $f_{\overline{ab}}$ for each of the $O(N)$ line segments in $A \cup B$ and returning the maximum value obtained by any $f_{\overline{ab}}$ function.

Theorem 4. *The* min-link *Hausdorff distance between line segment sets A and B in a simple polygon can be calculated in $O(kN + N^2 \log kN)$ time and $\min(O(kN), O(k + N^2))$ space.*[3]

Proof. Since any line segment intersects at most three faces of Suri's [18] shortest path map (see [6]), any function $f_{\overline{ab},\, \overline{cd}}$ is piecewise constant with at most three pieces. $f_{\overline{ab},\, \overline{cd}}$ can be calculated by computing a shortest path map from a line segment source $\overline{cd}$ in $O(k)$ time, finding the position of a in the shortest path map in $O(\log k)$ time, and walking through at most three faces of the shortest path map in $O(1)$ time. Note that after precomputing the shortest path map from $\overline{cd}$, any function $f_{\overline{ab},\, \overline{cd}}$ involving $\overline{cd}$ can be computed in $O(\log k)$ time.

$f_{\overline{ab}}$ is the lower envelope of $O(N)$ piecewise constant functions $f_{\overline{ab},\, \overline{cd}}$. Once all the piecewise constant functions are known, $f_{\overline{ab}}$ can be computed in $O(N \log N)$ time and $O(N)$ space by a planesweep. Computing a function $f_{\overline{ab}}$ for each of the $O(N)$ line segments in $A \cup B$ and returning the maximum value over all $f_{\overline{ab}}$ yields the Hausdorff distance.

The time bound follows by computing $O(N)$ shortest path maps in $O(kN)$ time [18], computing $O(N^2)$ $f_{\overline{ab},\, \overline{cd}}$ functions in $O(N^2 \log k)$ time, and computing $O(N)$ lower envelopes in $O(N^2 \log N)$ time. The space bound is $O(kN)$ if all shortest path maps are precomputed, or it is $O(k + N^2)$ if only one shortest path map is stored at a time while all $O(N^2)$ $f_{\overline{ab},\, \overline{cd}}$ functions are stored at once. □

---

[2]The runtime component $O(N^2 k)$ can be reduced to $O(N^2 \log k)$ if one is willing to work with approximate distances that differ by at most one link from the true min-link distance [17].

[3]Interestingly, this approach is a log factor slower than our corresponding Fréchet distance algorithm in section 3.4.

Theorem 5. *The* min-link *Hausdorff distance $\delta_H(A,B)$ between line segment sets A and B in a polygonal domain can be calculated* exactly *in $O(N^2k^2(\alpha(k)\log^2 k + \log Nk))$ time and $O(Nk^2)$ space. The Hausdorff distance can also be computed* approximately *to within 1 link of $\delta_H(A,B)$ in $O(N^2k^2\alpha(k)\log^2 k)$ time and $O(k^2)$ space, to within 2 links of $\delta_H(A,B)$ in $O(Nk^{\frac{7}{3}}\log^{3.11} k + N^2k)$ time and $O(k+N)$ space, and to within 3 links of $\delta_H(A,B)$ in $O(Nk^{\frac{7}{3}}\log^{3.11} k + N^2\log k)$ time and $O(k+N)$ space.*

Proof. The exact Hausdorff distance can be computed with the min-link shortest path map SPM($\overline{cd}$). For a given query point s in the polygonal domain, SPM($\overline{cd}$) returns $\min_{t\in\overline{cd}} d(s,t)$ in $O(\log k)$ time (see [17]). Hence, SPM($\overline{cd}$) can be used to compute the piecewise constant function $f_{\overline{ab},\,\overline{cd}}$ by following $\overline{ab}$ through SPM($\overline{cd}$). Although it would take $\Theta(k^4)$ time and space to explicitly construct SPM($\overline{cd}$) [17], we use Lemma 1 to construct $f_{\overline{ab},\,\overline{cd}}$ from SPM($\overline{cd}$) $\cap$ $\overline{ab}$ in $O(k^2\alpha(k)\log^2 k)$ time and $O(k^2)$ space.

The function $f_{\overline{ab}}$ is the lower envelope of $O(N)$ functions $f_{\overline{ab},\,\overline{cd}}$ (i.e., one function $f_{\overline{ab},\,\overline{cd}}$ for each choice of $\overline{cd}$), and these $O(N)$ functions can be constructed in $O(Nk^2\alpha(k)\log^2 k)$ time and $O(Nk^2)$ space. The lower envelope involves $O(Nk^2)$ constant functions and can be computed in $O(Nk^2\log Nk)$ time and $O(Nk^2)$ space via planesweep. By computing $f_{\overline{ab}}$ for each of the $O(N)$ line segments in $A\cup B$ and returning the maximum value on any of these functions, the Hausdorff distance can be computed exactly in $O(N^2k^2(\alpha(k)\log^2 k + \log Nk))$ time. Only one lower envelope is stored at a time, so $O(Nk^2)$ space is sufficient.

The approximation techniques for the Hausdorff distance are based on the idea of using a single value to approximate the set of all nearest neighbor distances between a pair of line segments $\overline{ab}\in A$ and $\overline{cd}\in B$. Depending on how the approximating value is calculated, we can obtain the *approximate* Hausdorff distance $\delta_H^a(A,B)$ to within 1, 2, or 3 links of the exact Hausdorff distance $\delta_H(A,B)$.

Let $i=\min_{s\in\overline{ab},\,t\in\overline{cd}} d(s,t)$. By the definition of min-link paths, *all* distances $d(s,t)$ for $s\in\overline{ab}$, $t\in\overline{cd}$ equal either i, $i+1$, or $i+2$. This follows because any path $\pi(s,t)$ that is composed of i links can be extended by two extra links into an upper bound for the min-link path $\pi(s',t')$ for any choice of $s'\in\overline{ab}$, $t'\in\overline{cd}$ (see Figure 3.2).

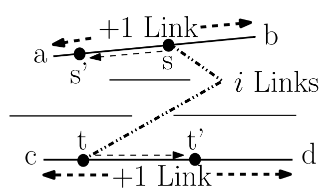

FIGURE 3.2. Approximate min-link paths between $\overline{ab}$ and $\overline{cd}$

Since every distance between a given pair of line segments equals i, $i+1$, or $i+2$, the approximating value $i+1$, must be within one link of all exact distances. By Lemma 1, the value of $i+1$ can be calculated for each of the $O(N^2)$ pairs of line segments in $O(k^2\alpha(k)\log^2 k)$ time and $O(k^2)$ space because i equals the minimum value on $f_{\overline{ab},\,\overline{cd}}$. The approximate Hausdorff distance can then be calculated using these distances in $O(N^2)$ additional time.

An approximation to within two links of $\delta_H(A,B)$ is even simpler because the exact value of i need not be known. Since any distance $d(s,t)$ between $\overline{ab}$ and $\overline{cd}$ must equal i, $i+1$, or $i+2$, any choice of $d(s,t)$ must be within two links of *all* exact distances between $\overline{ab}$ and $\overline{cd}$. $O(N^2)$ values of this type can be computed using $O(N)$ fixed-source structures due to Mitchell, Rote, and Woeginger [17]. Each structure can be built in $O(k^{\frac{7}{3}}\log^{3.11} k)$ time and $O(k)$ space from a fixed source point s and can answer queries between s and any point in the polygonal domain in $O(k)$ time. By storing only one of these fixed-source structures at a time, the approximate Hausdorff distance can be computed in $O(Nk^{\frac{7}{3}}\log^{3.11} k + N^2k)$ total time and $O(k+N)$ space [17].

The fixed-source structure of Mitchell, Rote, and Woeginger [17] also supports $O(\log k)$ time *approximate* distance queries that are accurate to within 1 link of the exact min-link distance. This allows the approximate Hausdorff distance to be computed in $O(Nk^{\frac{7}{3}} \log^{3.11} k + N^2 \log k)$ time and $O(k + N)$ space to within 3 links of the exact solution. □

3.4. **Min-Link Fréchet Distance in a Simple Polygon.** The Fréchet distance was defined in section 2 as a similarity metric that (unlike the Hausdorff distance) takes the continuity of curves into account. The *decision problem* is commonly used [4] to check whether the Fréchet distance is at most a given parameter $\varepsilon > 0$. Although the traditional Fréchet distance [4] assumes that the free space in a cell is convex, the framework of [4] for solving the decision problem still applies if this free space is x-monotone, y-monotone, and connected.

Lemma 2. *The free space in a cell $\mathcal{C}$ for the min-link Fréchet distance between polygonal curves A and B in a simple polygon is not convex in general, but the free space is x-monotone, y-monotone, and connected. Furthermore, the distance function $f_{s,\,\overline{cd}}$ for any horizontal or vertical line segment in $\mathcal{C}$ is $\downarrow\uparrow$-bitonic and has constant complexity.*

Proof. Figures 3.3a and 3.3b illustrate an example where the free space in $\mathcal{C}$ is not convex. However, the free space is x-monotone and y-monotone as shown next. The distance function $f_{s,\,\overline{cd}}$ for any horizontal or vertical line segment in $\mathcal{C}$ is defined by the min-link paths from a fixed-source point s to each point on a line segment $\overline{cd}$. Let $i = \min_{t \in \overline{cd}} d(s,t)$ be the length of the *shortest* min-link path from s to any point $t \in \overline{cd}$. It has been shown in [6] that all min-link paths from s to $\overline{cd}$ can be represented by at most three intervals on $\overline{cd}$ with respective lengths $i+1$, i, and $i+1$. This implies that the distance function $f_{s,\,\overline{cd}}$ is $\downarrow\uparrow$-bitonic and has constant complexity. This bitonicity plus the fact that free space consists of all distances less than or equal to ε ensures that the free space in $\mathcal{C}$ is x-monotone and y-monotone.

To see that the free space in $\mathcal{C}$ is *connected*, pick two arbitrary free space points (s,t) and (s',t') such that $d(s,t),\ d(s',t') \le \varepsilon$. We now show that a path through the free space always exists that connects these two points. Let $\pi(s,t)$ have the form $s, s_1, ..., t$ and $\pi(s',t')$ have the form $s', s_1', ..., t'$ (see Figures 3.3c and 3.3d). Let $\mathcal{T}$ be the region bounded by $\overline{s's}$, $\pi(s,t)$, $\overline{tt'}$, and $\pi(t',s')$. Although all min-link paths from $\rho \in \overline{ss'}$ to $\varrho \in \overline{tt'}$ need not lie entirely in $\mathcal{T}$, we shall see that it is possible to slide (s,t) through free space in $\mathcal{T}$ until it reaches (s',t').

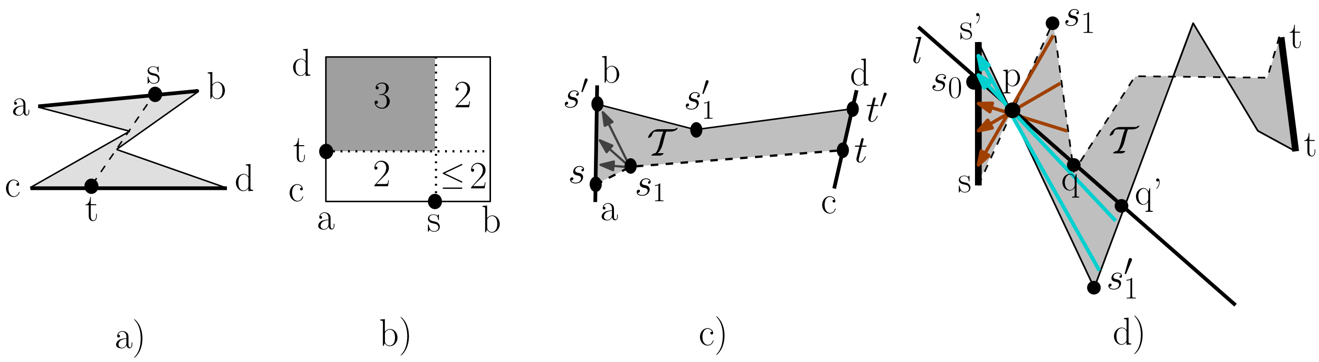

FIGURE 3.3. a) Two line segments in a simple polygon can produce b) a min-link free space cell with *non-convex* free space because the points (a,t) and (s,d) are in the (white) free space but there is no line of sight through the free space between these points. c,d) Min-link free space for a cell in a simple polygon is connected.

If s_1 can see every point on $\overline{ss'}$, then (s,t) can be slid through free space in $\mathcal{T}$ to (s',t) (see Figure 3.3c). Analogously, if s_1' can see every point on $\overline{ss'}$, then (s',t') can be safely slid to(s,t'). Otherwise, let $p = \overline{ss_1} \cap \overline{s's_1'}$ (see Figure 3.3d). There exists a line l and points q on $s_1, \ldots, t$ and q' on $s_1', \ldots, t'$

such that both q, q' lie on l and have line of sight to p. Let $s_0 = l \cap \overline{ss'}$. Min-link paths through free space in $\mathcal{T}$ can be constructed from any point on $\overline{s_0 s'}$ through p to a point on $s'_1, \ldots, q'$, and continuing to t'. Analogously, min-link paths through free space exist from any point on $\overline{ss_0}$ through p to a point on $s_1, \ldots, q$, and continuing to t. This shows that (s', t') can be safely slid to (s_0, t'), and (s, t) can be slid to (s_0, t). In all cases, the problem has been reduced to two points lying on a vertical line in the free space cell. Such a vertical line has a decreasing-increasing bitonic distance function. Hence, it is safe to translate these points vertically into each other, and the free space in $\mathcal{C}$ is connected. □

Theorem 6. *After precomputing all free space cell boundaries in $O(kN + N^2)$ time and $O(k + N^2)$ space, the* decision problem *for the min-link geodesic Fréchet distance between polygonal curves A and B in a simple polygon can be solved in $O(N^2)$ time and space.*

Proof. We precompute the cell boundaries of the free space diagram one row (or column) at a time. By contrast, the traditional approach [4] computes these boundaries one *cell* at a time. Each horizontal (or vertical) line segment $\mathcal{L}$ in the free space diagram is defined by a single fixed-source point s. Precompute SPM(s) and find the region in SPM(s) that contains the start point of the polygonal curve B in $O(k)$ time [18]. By following each line segment $b \in B$ through at most three regions of SPM(s) [6], all min-link distances from s to B can be computed in $O(k + N)$ total time and space.

By repeating this step for each of the $O(N)$ rows and columns that define cell boundaries in the free space diagram, we obtain $O(N \cdot (k + N))$ total preprocessing time. Since each cell stores at most three distances on each of its four edges, only constant storage is required for each of the $O(N^2)$ cells. By storing only one shortest path map at a time, the total space requirement for preprocessing is $O(k + N^2)$.

By Lemma 2, the free space in each cell is x-monotone, y-monotone, and connected. These properties allow the decision problem to be solved by computing free space on the cell boundaries and propagating reachability information in constant time per cell using the dynamic programming approach of Alt and Godau [4]. Only constant time per cell is needed because the free space on each cell's boundary has constant complexity (see Lemma 2). □

Theorem 7. *The min-link geodesic Fréchet distance in a simple polygon can be computed* exactly *in $O(kN + N^2)$ time and $O(k + N^2)$ space. It can also be computed with an* additive *approximation to within $\mathcal{B}$ links of optimal in $O(\frac{kN}{\mathcal{B}} + \frac{N^2}{\mathcal{B}^2})$ time and $O(k + \frac{N^2}{\mathcal{B}^2})$ space (for $\mathcal{B} \geq 2$).*

Proof. Theorem 6 allows the decision problem to be solved in $O(N^2)$ time and space after $O(kN+N^2)$ time and $O(k + N^2)$ space preprocessing. The traditional approach (see [4]) to solve the Fréchet optimization problem is to use parametric search which incurs a logarithmic overhead on top of the runtime for the decision problem. By contrast, we improve on the traditional approach by using an *approximation* algorithm to narrow the search space to $O(1)$ candidate values for the Fréchet distance. We then call the exact decision problem $O(1)$ times to compute the exact Fréchet distance.

Let a *bundle* be a group of $O(\mathcal{B}^2)$ connected free space cells, and let $\mathcal{D}$ be an exact distance for an arbitrary point in the bundle. We show next that for a *diamond-shaped* bundle of size $O(\mathcal{B}^2)$, $\mathcal{D}$ is within $\mathcal{B}$ links of all distances in the bundle.

Consider first a bundle containing only *one* free space cell and defined by $\overline{ab} \times \overline{cd}$. Let $i = \min_{s \in \overline{ab},\, t \in \overline{cd}} d(s, t)$. By the definition of min-link paths, *all* distances $d(s, t)$ for $s \in \overline{ab}$, $t \in \overline{cd}$ equal either i, $i + 1$, or $i + 2$ (see Figure 3.2). Hence, any distance $\mathcal{D}$ in this bundle is within two links of all distances in the bundle. Now suppose the bundle size is increased from one to five by adding four cells adjacent to the original cell at its left, top, right, and bottom edges (ignore diagonally adjacent cells). By a simple extension of Figure 3.2, $\mathcal{D}$ is approximate to within 3 links of all distances in these cells. By repeatedly applying this idea, a diamond-shaped bundle with x layers and $1 + 4 \cdot (1 + 2 + 3 + ... + x) \in O(x^2)$ free space cells exists such that any distance $\mathcal{D}$ in this bundle is within $x + 2$ links of all distances in the bundle. Hence, a representative distance $\mathcal{D}$ in a bundle can be made accurate to within $\mathcal{B}$ links (for $\mathcal{B} \geq 2$) by constructing a diamond-shaped bundle with $O(\mathcal{B}^2)$ cells.

For a bundle containing $O(\mathcal{B}^2)$ free space cells, $O(\frac{N^2}{\mathcal{B}^2})$ bundles are sufficient to cover the free space diagram. We organize these bundles into $O(\frac{N}{\mathcal{B}})$ columns and represent each bundle by a *node* and a representative distance in the center of the diamond-shaped bundle (see Figure 3.4b). $O(\frac{N}{\mathcal{B}})$ representative distances for the bundle nodes in a single column can be computed using a single shortest path map. Traditional point location would calculate these distances in $O(k + \frac{N}{\mathcal{B}} \log k)$ time per column (including the construction of the shortest path map) [18]; however, we can improve this to $O(k + \frac{N}{\mathcal{B}})$ time as follows.

Notice that if all bundle nodes are projected onto the vertical axis of the free space diagram (see Figure 3.4b), there are only $O(\frac{N}{\mathcal{B}})$ unique node positions that correspond to a set of $O(\frac{N}{\mathcal{B}})$ points on the polygonal curve B. By Observation 1, a polygonal path of length $O(k + \frac{N}{\mathcal{B}})$ that visits all of these points can be constructed by a one-time preprocessing step in $O(k + \frac{N}{\mathcal{B}} \log k)$ time and $O(k + \frac{N}{\mathcal{B}})$ space. This path is useful because any line segment intersects at most three faces of a min-link shortest path map for a simple polygon [6]. Hence, a polygonal path of length $O(k + \frac{N}{\mathcal{B}})$ can be traced through a (triangulated) shortest path map in $O(k + \frac{N}{\mathcal{B}})$ time. This allows all representative distances for the $O(\frac{N}{\mathcal{B}})$ columns of bundles to be constructed in $O(\frac{N}{\mathcal{B}} \cdot (k + \frac{N}{\mathcal{B}}))$ time and $O(k + \frac{N^2}{\mathcal{B}^2})$ space.

After defining representative distances for the bundles, a directed acyclic graph with $O(\frac{N^2}{\mathcal{B}^2})$ nodes and edges can be used to represent all monotone paths through the bundles (see Figure 3.4c). Note that the number of edges is bounded because the out-degree of any node is at most three. A breadth first search in $O(\frac{N^2}{\mathcal{B}^2})$ time over this graph yields an *additive* approximation for the Fréchet distance that is within $\mathcal{B}$ links of optimal.

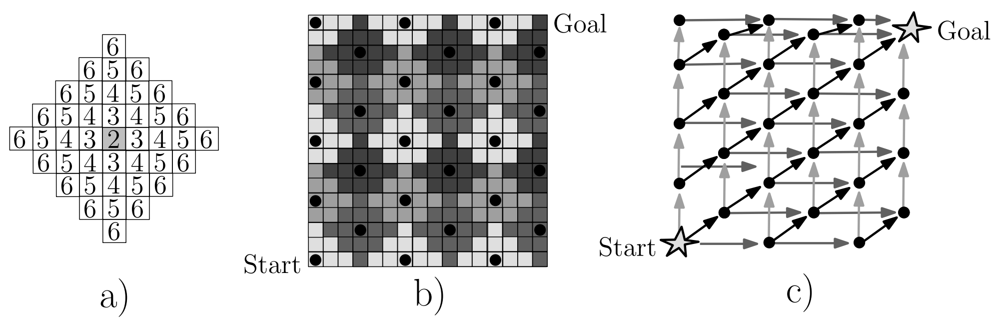

FIGURE 3.4. a) A bundle of free space cells, where each cell in the bundle is illustrated with its additive approximation from a representative distance $\mathcal{D}$ in the shaded node. b) The free space diagram can be covered by $O(\frac{N^2}{\mathcal{B}^2})$ bundles. c) The directed acyclic graph enforces path monotonicity between bundles, allowing the approximate Fréchet distance to be calculated via breadth first search.

The *exact* Fréchet distance can be computed by first applying the approximation algorithm with a constant bundle size (i.e., $\mathcal{B} = 2$) to achieve an approximation to within $O(1)$ links of optimal. The *exact* decision problem can then be executed $O(1)$ times to compute the Fréchet distance in $O(N^2)$ additional time. □

Corollary 1. *The traditional (no obstacles) Fréchet distance between polygonal curves A and B in $\mathbb{R}^d$ can be computed with an* additive $\mathcal{B} \cdot l_{\max}$ *approximation in $O(\frac{N^2}{\mathcal{B}^2})$ time and space, where $l_{\max}$ is the length of the longest line segment in $A \cup B$.*

Proof. The $O(\frac{N^2}{\mathcal{B}^2})$ bundles and directed acyclic graph can be computed in $O(\frac{N^2}{\mathcal{B}^2})$ time because each bundle's representative distance can be computed in $O(1)$ time. The $\mathcal{B} \cdot l_{\max}$ bound for the additive approximation holds because adding another "ith-layer" of $4i$ cells to the bundle can increase the true Euclidean distance of any point in the bundle by at most $l_{\max}$ (see Figures 3.2 and 3.4a). □

The bundles technique is novel because (1) it supports the first *sub-quadratic* approximation algorithm for the traditional or min-link Fréchet distance and (2) it allows the *exact* min-link Fréchet optimization problem to be computed in the same asymptotic time and space as the decision problem.

3.5. **Min-Link Fréchet Distance in a Polygonal Domain.** The Fréchet distance is more difficult to compute in a polygonal domain than in a simple polygon because the free space inside a cell need neither be monotone nor connected (see Figure 3.5).

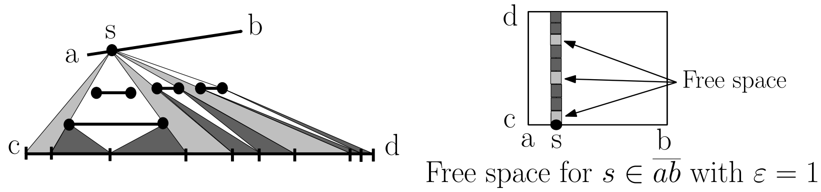

FIGURE 3.5. Free space in a polygonal domain need neither be monotone nor connected.

Lemma 3. *The free space diagram for the min-link Fréchet distance in a polygonal domain has* $\Omega(N^2k^4)$ *complexity.*

Proof. Figure 3.6 illustrates a single free space cell $\overline{ab} \times \overline{cd}$ with $\Omega(k^4)$ complexity. In Figure 3.6a, $\overline{ab}$ and $\overline{cd}$ are enclosed in rectangles with *point* openings. $\Omega(k^2)$ line of sight edges can be drawn between pairs of point openings. Let the intersections of these $\Omega(k^2)$ edges with $\overline{ab}$ be $a_1, ..., a_{\Omega(k^2)}$ and the intersections with $\overline{cd}$ be $c_1, ..., c_{\Omega(k^2)}$ (see Figure 3.6a). For any line of sight edge $\overline{a_ic_j}$, $d(a_i, c_j) = 1$ and $d(a_i, t) = 2$ for all $t \in \overline{cd},\ t \neq c_j$. Now choose $s \in \overline{a_ia_{i+1}},\ s \neq a_i, a_{i+1}$. Notice that $d(s, c_j) = 2$ for any $c_1, ..., c_{\Omega(k^2)}$ and that $d(s, t') = 3$ for all $t' \in \overline{cd},\ t' \neq c_1, ..., c_{\Omega(k^2)}$. Hence, $a_1, ..., a_{\Omega(k^2)}$ and $c_1, ..., c_{\Omega(k^2)}$ define $\Omega(k^2)$ horizontal and vertical lines in $\overline{ab} \times \overline{cd}$ such that all distances on these lines are at most 2, and all other distances are 3. By choosing $\varepsilon = 2$, the free space is the union of $\Omega(k^2)$ horizontal and vertical lines and has $\Omega(k^4)$ total complexity (see Figure 3.6b).

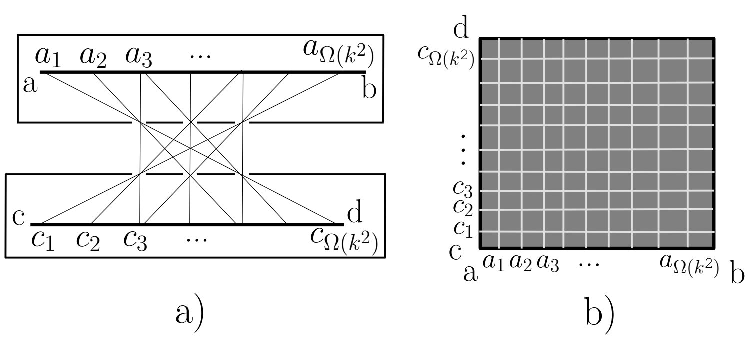

FIGURE 3.6. a) Two line segments in a min-link polygonal domain can define b) a single free space cell with $\Omega(k^4)$ complexity.

Define the polygonal curves by their endpoints as $A = \{a, b, a, b, ..., a, b\}$ and $B = \{c, d, c, d, ..., c, d\}$. Since the $\frac{N^2}{4}$ cells defined by $\overline{ab} \times \overline{cd}$ each have $\Omega(k^4)$ complexity, the free space diagram has $\Omega(N^2k^4)$ complexity. □

Theorem 8. *The min-link Fréchet distance* $\delta_F(A, B)$ *between polygonal curves* A *and* B *in a polygonal domain can be calculated* exactly *in* $O(N^2k^7 \log kN)$ *time and* $O(N^2k^7)$ *space. The Fréchet distance can also be computed* approximately *to within 1 link of* $\delta_F(A, B)$ *in* $O(N^2k^2\alpha(k)\log^2 k)$ *time and* $O(k^2 + N^2)$ *space, to within 2 links of* $\delta_F(A, B)$ *in* $O(Nk^{\frac{7}{3}} \log^{3.11} k + N^2k)$ *time and* $O(k + N^2)$ *space, and to within 3 links of* $\delta_F(A, B)$ *in* $O(Nk^{\frac{7}{3}} \log^{3.11} k + N^2 \log k)$ *time and* $O(k + N^2)$ *space.*

Proof. $\delta_F(A,B)$ can be computed *exactly* as follows. For each of the $O(N^2)$ free space cells, precompute SPM($\overline{ab}$,$\overline{cd}$). This process takes $O(N^2k^7)$ time and space by Theorem 2. By construction, the $O(N^2k^7)$ faces in these shortest path maps form a partition of the free space diagram, and each face has the same min-link distance throughout its interior. The Fréchet decision problem (cf. section 2) can be solved by using a planesweep to propagate reachability information through the free space diagram. Simply sort the leftmost and rightmost endpoints of the $O(N^2k^7)$ faces and propagate reachability by sweeping a vertical line L from left to right while maintaining *free space* and *reachable space* information in $O(N^2k^7 \log kN)$ time. Instead of using parametric search, any of the below approximation algorithms can be applied to quickly determine the Fréchet distance to within γ links of the true value. The exact decision problem can then be executed γ times to determine $\delta_F(A,B)$ in $O(N^2k^7 \log kN)$ time.

$\delta_F(A,B)$ can be computed *approximately* as follows. Let $\mathcal{C}$ be an (exact) free space cell defined by the line segments $\overline{ab} \in A$ and $\overline{cd} \in B$, and let $i = \min_{s\in\overline{ab},\ t\in\overline{cd}} d(s,t)$. All distances in the free space cell $\mathcal{C}$ are either i, $i+1$, or $i+2$ (see the proof of Theorem 5), and this allows defining an *approximate* free space cell $\mathcal{C}^a$ such that all distances in $\mathcal{C}^a$ equal a single value. For any $\varepsilon \geq 0$, $\mathcal{C}^a$ has constant complexity and is composed entirely of free space or constrained space.

Suppose an approximate free space diagram is computed such that each approximate distance $d^a(s,t)$ is within γ links of the exact distance $d(s,t)$. $d^a(s,t) \geq d(s,t) - \gamma$ guarantees that the approximate free space created for a value ε is a subset of the exact free space created for a value $\varepsilon - \gamma$. Similarly, $d(s,t) + \gamma \geq d^a(s,t)$ guarantees that the exact free space created for a value of $\varepsilon + \gamma$ is a subset of the approximate free space created for ε. Hence, the approximate Fréchet distance is within γ links of the exact Fréchet distance.

To approximate the Fréchet distance to within one link of $\delta_F(A,B)$, compute a representative value $i+1$ for each of the $O(N^2)$ cells. This can be done as follows (cf. Theorem 5). For a given query point s in a polygonal domain, SPM($\overline{cd}$) returns $\min_{t\in\overline{cd}} d(s,t)$ in $O(\log k)$ time (see [17]). Hence, $i = \min_{s\in\overline{ab},\ t\in\overline{cd}} d(s,t)$ is the *shortest* distance defined by SPM($\overline{cd}$) $\cap$ $\overline{ab}$. Although it would take $O(k^4)$ time and space to explicitly construct SPM($\overline{cd}$) [17], we use Lemma 1 to construct SPM($\overline{cd}$) $\cap$ $\overline{ab}$ in $O(k^2\alpha(k)\log^2 k)$ time and $O(k^2)$ space. Repeating this process for each cell takes $O(N^2k^2\alpha(k)\log^2 k)$ total time and $O(k^2+N^2)$ total space. Once a representative value is known for each free space cell, the approximate Fréchet distance can be solved in $O(N^2)$ time using breadth first search (cf. Theorem 7).

To approximate the Fréchet distance to within two links of $\delta_F(A,B)$, let the representative value for an approximate cell $\mathcal{C}^a$ equal any exact distance $d(s,t)$ for $s \in \overline{ab}$ and $t \in \overline{cd}$. By precomputing $O(N)$ fixed-source structures in $O(Nk^{\frac{7}{3}} \log^{3.11} k)$ time, the $O(N^2)$ representative values can be computed in $O(N^2k)$ total time [17]. Each fixed-source structure requires $O(k)$ space and only one of these structures needs to be stored at a time, so $O(k+N^2)$ space is sufficient. The fixed-source structure also supports finding an *approximate* representative value that is accurate to within one link of an exact distance in $O(\log k)$ time. This technique yields an approximate Fréchet distance to within three links of $\delta_F(A,B)$. ☐

4. CONCLUSION

We have developed a new type of min-link shortest path map that can answer queries from any source point on a *line segment*. This technique is more versatile than the traditional *point* source shortest path map, and we apply this structure to solve the min-link Fréchet distance in a polygonal domain. In addition to exact algorithms, we also present approximations for the min-link Hausdorff and Fréchet distance that are accurate to within 1, 2, or 3 links of optimal. We are currently working on finding highly accurate approximations for Hausdorff and Fréchet distance problems under the Euclidean metric. Recent advances in this direction have approximated a variant of the Fréchet distance called the *discrete* Fréchet distance [7] and have used the Fréchet distance to approximately simplify a curve [1].

REFERENCES

[1] P. K. Agarwal, S. Har-Peled, N. Mustafa, and Y. Wang. Near-linear time approximation algorithms for curve simplification in two and three-dimensions. *Algorithmica*, 42(3-4):203–221, 2005.

[2] P. K. Agarwal and M. Sharir. *Davenport–Schinzel Sequences and Their Geometric Applications*, pages 1–47. Handbook of Computational Geometry, Elsevier, Amsterdam, 2000.

[3] H. Alt, P. Braß, M. Godau, C. Knauer, and C. Wenk. Computing the Hausdorff distance of geometric patterns and shapes. In *Discrete and Computational Geometry.*, volume 25 of *Algorithms and Combinatorics*, pages 65–76. Springer, Berlin, 2003. Special Issue: The Goodman-Pollack-Festschrift (B. Aronov, S. Basu, J. Pach, M. Sharir eds.).

[4] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, 5:75–91, 1995.

[5] H. Alt, C. Knauer, and C. Wenk. Comparison of distance measures for planar curves. *Algorithmica*, 38(1):45–58, 2003.

[6] E. M. Arkin, J. S. B. Mitchell, and S. Suri. Optimal link path queries in a simple polygon. *SODA: 3rd ACM-SIAM Symposium on Discrete Algorithms*, pages 269–279, 1992.

[7] B. Aronov, S. Har-Peled, C. Knauer, Y. Wang, and C. Wenk. Fréchet distance for curves, revisited. *14th Annual European Symposium on Algorithms (ESA)*, pages 52–63, 2006.

[8] E. W. Chambers, É. C. de Verdière, J. Erickson, S. Lazard, F. Lazarus, and S. Thite. Walking your dog in the woods in polynomial time. *SoCG: 24th Symposium on Computational Geometry*, pages 101–109, 2008.

[9] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete and Computational Geometry*, 6:485–524, 1991.

[10] Y. Chiang and J. S. B. Mitchell. Two-point Euclidean shortest path queries in the plane. *SODA: 10th ACM-SIAM Symposium on Discrete Algorithms*, pages 215–224, 1999.

[11] A. F. Cook IV and C. Wenk. Geodesic Fréchet distance inside a simple polygon. *Proceedings of the 25th International Symposium on Theoretical Aspects of Computer Science (STACS), Bordeaux, France*, 2008.

[12] A. Efrat, L. J. Guibas, S. Har-Peled, D. C. Lin, J. S. B. Mitchell, and T. M. Murali. Sweeping simple polygons with a chain of guards. *SODA: 11th ACM-SIAM Symposium on Discrete Algorithms*, pages 927–936, 2000.

[13] 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*, 2:209–233, 1987.

[14] A. Maheshwari, J.-R. Sack, and H. N. Djidjev. Link distance problems. *Handbook of Computational Geometry*, 1999.

[15] A. Maheshwari and J. Yi. On computing Fréchet distance of two paths on a convex polyhedron. *European Workshop on Computational Geometry (EWCG)*, pages 41–4, 2005.

[16] J. S. B. Mitchell. Geometric shortest paths and network optimization. *Handbook of Computational Geometry*, 1998.

[17] J. S. B. Mitchell, G. Rote, and G. J. Woeginger. Minimum-link paths among obstacles in the plane. *SoCG: 6th Symposium on Computational Geometry*, pages 63–72, 1990.

[18] S. Suri. A linear time algorithm for minimum link paths inside a simple polygon. *Computer Vision and Graphical Image Processing (CVGIP)*, 35(1):99–110, July 1986.

[19] S. Suri and J. O'Rourke. Worst-case optimal algorithms for constructing visibility polygons with holes. *SoCG: 2nd Symposium on Computational Geometry*, pages 14–23, 1986.